

DEVELOPMENT OF A COMPUTATIONAL IMAGE SENSOR WITH APPLICATIONS IN INTEGRATED SENSING AND PROCESSING

A Dissertation
Presented to
The Academic Faculty

By

Ryan W. Robucci

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering



School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2009

DEVELOPMENT OF A COMPUTATIONAL IMAGE SENSOR WITH APPLICATIONS IN INTEGRATED SENSING AND PROCESSING

Approved by:

Dr. Paul E. Hasler, Advisor
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. Justin Romberg
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. David V. Anderson
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. Mark Smith
Professor, Department of Communications Sys-
tems at the Kungliga Tekniska Högskolan
Swedish Royal Institute of Technology
Stockholm, Sweden

Dr. Maysam Ghovanloo
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Date Approved: March 2009

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
SUMMARY	ix
CHAPTER 1 INTEGRATED REPROGRAMMABLE ANALOG IMAGE PROCESSING	2
1.1 Reprogrammable Analog Hardware	3
1.2 Mixed-Mode Distributed Processing	5
CHAPTER 2 CMOS IMAGERS	7
2.1 Basic Photoreceptor Circuits	7
2.2 Active Pixel Sensor (APS) Imagers	10
2.3 High Dynamic Range Imaging Techniques	13
2.4 Focal-Plane Processing	14
2.5 Integrated Sensing and Processing and Intellegent ICs	16
CHAPTER 3 SUBTHRESHOLD CONDUCTION AND FLOATING-GATE TRANSISTORS	18
3.1 Subthreshold Transistor Modeling	18
3.2 Subthreshold Floating-Gate Transistor Operation	20
3.3 Reprogrammable Analog Floating-Gate Transistors	20
CHAPTER 4 COMPUTATIONAL FOCAL PLANE	24
4.1 Image Processing Using Matrix Operations	24
4.1.1 Signal Processing with Matrix Operations	26
4.1.2 Two-Dimensional Image Processing with Separable Transforms	27
4.2 Computational Pixel Operation and Characterization	28
4.3 Validation of Voltage-Light Multiplication	30
CHAPTER 5 COMPUTATIONAL SENSING SYSTEM ARCHITECTURE	35
5.1 Computational Pixel Tile for In-Pixel A-Matrix Multiplication	38
5.2 Random Access Analog Memory for the A-Matrix	38
5.3 Current Sensing and Processing for B-Matrix Multiplication	39
5.4 Archetecture Improvements	45
CHAPTER 6 SENSING AND PROCESSING LOW CURRENT, WIDE DYNAMIC RANGE SIGNALS	49
6.1 Programmable Subthreshold Current Mirroring	50
6.2 Logarithmic Transimpedance Amplifiers	54
6.2.1 Noise	56
6.3 Bi-Directional Compressive Transimpedance Amplifier	61

CHAPTER 7	MISMATCH AND OFFSET REMOVAL	65
7.1	Pixel Array Characteristics and Mismatch	65
7.2	Pixel Plane Design for Reduced Parasitics	75
7.3	Offset Removal	75
7.4	Double Reading	79
7.5	Dual-Slope Integration	83
CHAPTER 8	APPLICATION IN COMPRESSIVE SENSING	89
8.1	Transform Image Sensor	91
8.2	Sensing with Decorrelated Basis Functions	93
8.3	Results	94
8.4	Conclusion	98
CHAPTER 9	COMPUTATIONAL RESULTS	99
CHAPTER 10	CONCLUSION	104
10.1	Detailed Computational Pixel Investigation	104
10.2	Pixel Plane Mismatches, Offsets, and Error-Correction Modeling	105
10.3	New Architectures Enabling Increased Functionality and Performance	106
10.4	Reduced Parasitic Pixel and Pixel-Plane	106
10.5	High-Speed Analog Memory	107
10.6	Wide Range Current Sensing and Processing	107
10.7	Physical System Implementation and Applications	108
REFERENCES		110

LIST OF TABLES

Table 7.1	Pixel statistics extracted from a pixel array	69
-----------	---	----

LIST OF FIGURES

Figure 1.1	Reconfigurable transform imager system	4
Figure 2.1	Photo diode	9
Figure 2.2	Basic photoreceptor circuits	11
Figure 2.3	Active Pixel Sensor (APS) array	12
Figure 2.4	Measured APS pixel operation.	13
Figure 2.5	Architecture of traditional vs. focal plane processing.	15
Figure 2.6	Information Sensor.	17
Figure 3.1	Reprogrammable floating-gate transistor	19
Figure 3.2	Hot-electron injection.	21
Figure 3.3	Floating-gate array programming.	22
Figure 4.1	Reprogrammable computational image sensor	25
Figure 4.2	Differential pixel	29
Figure 4.3	Pixel characterization	31
Figure 4.4	Pixel currents with varying intensity	33
Figure 4.5	Photosensor tail current as a function of light intensity controlled using light absorption filters	33
Figure 4.6	The transconductance of the differential amplifier related to light intensity and saturation current	34
Figure 5.1	Computational imager sensor system level diagram	36
Figure 5.2	Die photograph of 256x256 imager	36
Figure 5.3	Computational imager sensor separable transform operation	37
Figure 5.4	Pixel tile	39
Figure 5.5	Random access analog float-gate biased memory	40
Figure 5.6	Fully differential 16x16 vector matrix multiplier	41
Figure 5.7	Differential to single-ended I-V converter	42

Figure 5.8	Multiplicative response of a programmable current mirror	44
Figure 5.9	Preliminary image results of a parking lot and garage from a window view	45
Figure 5.10	Computational imager sensor system level diagram	46
Figure 5.11	Pixel output logarithmic amplifiers	47
Figure 5.12	Newest image sensor IC eie photo	48
Figure 6.1	Current mirrors	51
Figure 6.2	Source to gate coupling	53
Figure 6.3	Logarithmic transimpedance amplifier topologies	54
Figure 6.4	Simple I-V	55
Figure 6.5	Logarithmic transimpedance amplifier noise sources	56
Figure 6.6	Logarithmic amplifier feedback element gain	59
Figure 6.7	Dynamic amplifier	61
Figure 6.8	Bidirectional I-Vs	62
Figure 7.1	Current offsets showing large column striations (column offsets)	66
Figure 7.2	Average column voltage offsets and column current offsets	67
Figure 7.3	Gain mismatch	68
Figure 7.4	Kappa mismatch	68
Figure 7.5	Linear range	68
Figure 7.6	Voltage offsets	69
Figure 7.7	Voltage as a function of position, showing a mostly random distribu- tion of voltage offset	70
Figure 7.8	Overlapping linear ranges	71
Figure 7.9	Adjacent pixel mismatch	73
Figure 7.10	Edge effects of two different imager layouts with the same pixel design but different peripheral circuitry	74
Figure 7.11	Pixels with leakage currents.	76
Figure 7.12	Mismatch and parasitic current removal using chopper stablization . .	78

Figure 7.13	Images of mismatch removal on 256x256 imager	78
Figure 7.14	Double reading	79
Figure 7.15	Results while reading a raw image	82
Figure 7.16	Results while reading an image using an identity matrix transform in the linear region with “off” blocks set to 0 V common mode	84
Figure 7.17	DCT offset removal results using a zero matrix read	85
Figure 7.18	Mismatch removal on 256x256 imager	86
Figure 7.19	Switch imager design for double reading and dual slope integration . .	87
Figure 7.20	Dual slope integration voltage outputs	88
Figure 7.21	Dual slope integration vs. double reading results	88
Figure 8.1	Compressive Sensing system design	89
Figure 8.2	Separable transform image sensor hardware platform	91
Figure 8.3	Block matrix computation performed in the analog domain	92
Figure 8.4	DCT and noiselet basis functions	95
Figure 8.5	PSNR of reconstruction vs. percentage of used transform coefficients .	96
Figure 8.6	Reconstruction results using DCT and noiselet basis sets with various compression levels	97
Figure 9.1	System error derivation	100
Figure 9.2	Identity data, DCT data, and error image.	101
Figure 9.3	Histograms of reference data and reconstructed data	102
Figure 9.4	Error energy loss in compression	103
Figure 9.5	Error loss with non-standard compression	103

SUMMARY

The objective of this research was to build a reprogrammable computational imager utilizing on-chip analog computations for the purpose of studying the capabilities of integrated sensing and processing. Unlike conventional imaging systems, which acquire image data and perform calculations on it, this system tightly integrates the computation and sensing into one process. This allows the exploration of intelligent and efficient sensory and processing. The IC architecture and circuit designs have focused on wide dynamic range signals. The fundamental computation performed is a separable two-dimensional transform. This allows various operations, including block transformations and separable convolutions. The operations are reprogrammable and utilize analog memory and processing along with digital control. The random access to both the image plane and the computational operations allows for intraframe transform variations creating a hardware foundation for dynamic sampling and computation. One can also capture scenes with non-uniform resolution. Advantages, including utilization of feedback from processing to sensing and extensions of the technology including support for wavelets and larger transforms are also explored.

In the first chapter of this thesis, I discuss the integration of a reprogrammable, analog, signal processing technology onto image sensor ICs. The advantages of having computational hardware integrated with sensors is discussed. In addition, the advantages of having reconfigurable hardware for logistical and algorithmic purposes is given. Our analog processing circuitry innately provides reprogrammability and presents an obvious competition to reprogrammable digital counterparts.

In the second chapter, I give a selective overview of CMOS imagers and the technologies leading up to the work here. The inherent challenges in image sensing to handle large data quantities and wide-dynamic-range signals is discussed. Previous work in the integration of processing circuitry, especially focal-plane processing, is given. As demonstrated,

analog processing circuitry is compact enough to be integrated with sensing circuitry and is well-suited for processing the physical-based, analog signals from the sensing circuitry. It can provide the needed level of up-front processing that can reduce data throughput through the rest of the system.

In the third chapter, I introduce subthreshold conduction in field-effect transistors and floating-gate transistors. The properties of transistors operating in the subthreshold current regime, as opposed to the above threshold regime, offer exponential-based current-voltage relationships that are advantageous for implementing computations. Reprogrammable floating-gate transistors, which serve as part of the foundation of our analog processing technology, are presented.

In the fourth chapter, I present the computational, focal plane used for image sensing. It provides a critical computational ability, but avoids large and complex processing circuitry that can reduce image sensitivity. The functionality of the pixel array is discussed as well as the experimental testing and characterization of it.

In the fifth chapter, I discuss the architecture of the imager. The individual subsystems of the imager are each discussed. The mathematical function and circuit implementation of each is discussed along with the issues critical to their design.

In the sixth chapter, I elaborate on the circuit design challenges when processing with small currents that vary over multiple orders of magnitude. Tunable current mirrors, which provide the foundation for computation capabilities are heavily discussed. To extend the speed of the operations and provide the ability to process small input currents, logarithmic transimpedance amplifiers are given. The analytical relationship of power consumption to dynamic range is given along with circuits that improve power consumption by dynamically changing gain. Limitations of these circuits in terms of distortion and noise are given. Also, a circuit that can convert a bidirectional current input into a logarithmic representation is presented.

In the seventh chapter, I discuss the limitations of the computational pixel plane, including mismatch characteristics. The aggregate effects of mismatch are studied and the process of removing the dominant components of error is given.

In the eighth chapter, I discuss an application, compressive sensing, and how the unique capabilities presented by the computational image sensor are suited for it. Compressive sensing takes advantage of a small amount of a priori knowledge to greatly reduce the number of samples of a signal that must be taken. The idea is strongly related to the concept reducing data in the front-end of a system to reduce component throughputs. Even though this imager was not explicitly designed for compressive sensing, the flexibility and reprogrammability built into the design allowed the use of it in the compressive sensing framework. This exemplifies the advantages of reconfigurable and reprogrammable systems.

In the ninth chapter, I analyzed the resulting images from the hardware. Unlike traditional images sensors, the outputs of this IC are computational results, not raw data. However, this IC can be configured to output raw images as well, allowing a comparison between the two. In particular, a discrete cosine transform (DCT) output is compared to a standard image output.

In the tenth chapter, I summarize the contributions of my thesis work.

CHAPTER 1

INTEGRATED REPROGRAMMABLE ANALOG IMAGE PROCESSING

Vision systems must transduce, transfer, and process large quantities of visual information. Typically, the front-end of this sequence falls naturally in the analog domain, while the back-end processing is done mostly in the context of digital processing. The evolution of digital signal processing (DSP) theory and powerful digital hardware has created a wealth of opportunities to exploit application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), and digital signal processors (DSPs). However, analog circuits have a history of real-time computation and are particularly well suited for sensory interface applications.

Unfortunately, large analog systems have been difficult to implement, primarily because they suffer from the effects of mismatch in circuit components. Handling these mismatches in a systematic manner throughout the system design is difficult. Layout techniques aimed at reducing mismatch require more area and higher power consumption. Another shortcoming of analog systems has been the lack of reprogrammability or memory that exists in their digital counterparts. Now, as a developing technology, analog floating-gate transistor techniques address many of these issues, providing tunability to analog circuits that allow for matching, reprogrammability, memory, and computation. Thus, the abilities of analog circuitry to naturally handle sensory signals and perform certain low-power computations can be harnessed in large-scale, reprogrammable integrated systems.

The objective of this research is to investigate a use of reprogrammable, analog computational technology integrated with sensing hardware by building and testing a reprogrammable, analog processing imager. The goal is to obtain an understanding of some of the possibilities and challenges in low-current, wide-range sensing and real-world signal processing with reprogrammable analog circuitry. The imager IC is integrated into a

mixed-signal processing system to evaluate discrete cosine transforms (DCTs) and other algorithms developed for dynamic-resolution sensing and classification. The programmable computation and interface on the imager IC provides the ability to eventually create unique feedback paths from computations to front-end computational sensory.

Feedback from processing to sensing can serve as the foundation for a variety of adaptable systems to be developed in the future. These adaptable integrated circuits and systems will selectively and intelligently acquire visual information instead of simply piping along large amounts of pixel data. The development of reprogrammable or reconfigurable components will drive the development of high-level algorithms and robust systems. Possible end-user applications of efficient, intelligent sensors include airport surveillance and security, unmanned aerial vehicles (UAVs), low-power remote field sensors, mobile personal devices, traffic monitoring, human-computer interfacing, face recognition, biometrics, and assistive devices for the visually impaired.

As discussed in [1–3], the basic architecture used in this research results in large power savings by moving digital processing into the low-power analog domain while maintaining reconfigurability. Figure 1.1 illustrates this concept using JPEG compression. The dashed lines encompass reconfigurable components. In Figure 1.1(a), an FPGA receives digital signals from analog-to-digital converters and performs all of the needed computations in the digital domain. Figure 1.1(a) shows the DCT operation being performed on-chip in the analog domain. The computational elements remain reconfigurable, utilizing analog floating-gate technology.

1.1 Reprogrammable Analog Hardware

Reprogrammability in hardware systems has proved to be a crucial asset in technology development chains, starting from research and going all the way to marketable products. Reprogrammable platforms enable rapid research and development, since several iterations of designs can be tested and analyzed in a short time compared to the weeks or months that

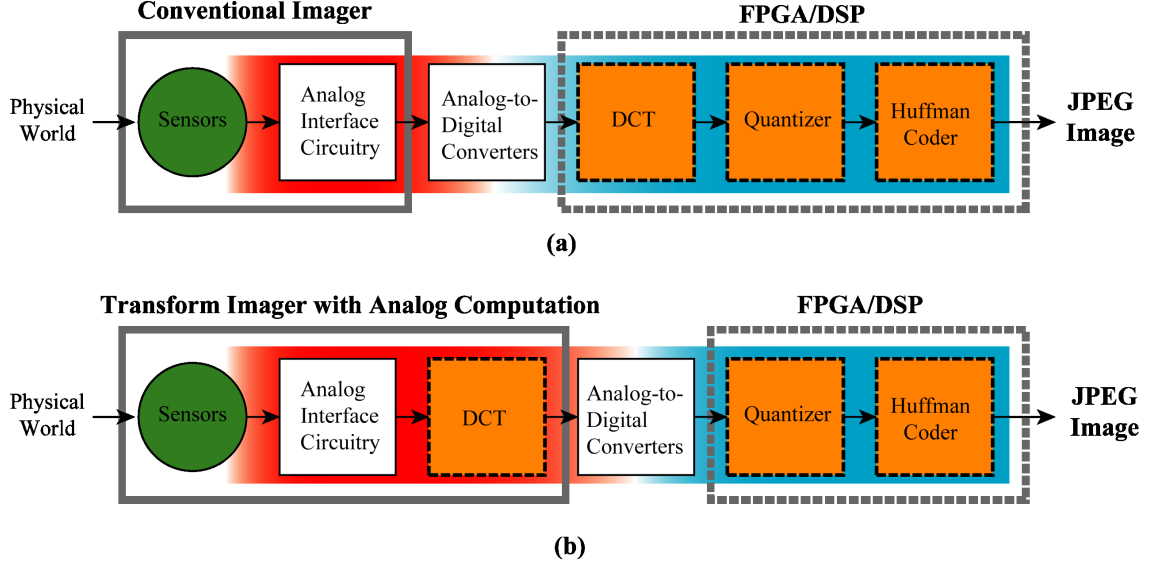


Figure 1.1. Reconfigurable transform imager system with (a) digital processing and (b) mixed-signal processing.

IC fabrications can take. Though they are not optimized for every application, their tendency to be well characterized before application-specific development commences leads to effective high-level optimizations. These advantages, along with mass-quantity production, offer low initial development costs for application developers. Furthermore, in-field reprogrammability allows bug fixes, performance tweaking, and feature set expansion after the product has been deployed. Most of these flexibilities are presently associated with digital systems. Analog reprogrammable technology can be used to make low-power analog systems versatile so that the same benefits can be realized in analog processing.

Even though reprogrammability comes at a considerable cost in digital systems, programmable parameters are an inherent part of analog floating-gate hardware. Reprogrammable analog components have been shown to be very efficient power when compared to their digital counterparts. Therefore, in a reprogrammable sensing and processing system, it is expected that power saving can be achieved when computational analog components are

integrated into the required front-end analog subsystem. This can reduce the digital subsystem power consumption and, consequently, overall system power efficiency. In [4], reprogrammable floating-gate vector-matrix multiplier cells were shown with 3.7 GMAC/s/mW efficiency ($\text{MAC/s/mW} = 1$ multiply accumulate per second per milliwat). Low-power DSPs operate with efficiencies at about 10 MMAC/s/mW [5]. In [6], an FPGA performing JPEG compression on a 104×128 video at 25 fps saved 146 mW by removing the two-dimensional discrete cosine transform (2-D DCT) operations. In contrast to the reprogrammable digital systems, a custom, low-power 2-D DCT ASIC, which was designed on a process similar to the one used for some of the imagers designed in this work, performs an 8×8 2-D DCT. That operation is the equivalent of two matrix multiplications, and is performed at 2.34 million blocks per second using 10 mW [7] of power. This is the equivalent of 46.8 million 8-by-8 matrix multiplications per second per milliwat, or 24 GMAC/s/mW. However, this custom design achieves these results through several optimizations and computational reductions targeted specifically for a sole computation with completely fixed values for the multiplications. This custom IC would perform the 2-D DCT operations the FPGA did with only 22 uW. So, it is obvious that digital reprogrammability is expensive. This cost presents an opportunity for power savings using reprogrammable analog circuitry and thus a motivation to explore the integration of reprogrammable analog computational abilities into analog sensing circuitry.

1.2 Mixed-Mode Distributed Processing

With the ability to build tunable analog circuits, one can create smaller analog components and levels of on-chip integration that approach those in digital systems. The compact size of analog floating-gate computational elements enables the creation of parallel, distributed analog-processing structures that minimize transmission power consumption. However, digital systems maintain advantages in high signal-to-noise-ratio (SNR) applications. This

is because cost increases logarithmically with SNR in digital systems and it increases linearly in analog systems (but only at lower SNR). At low SNR, analog computations can be more efficient than digital computations[8].

With regard to image processing, several bits are typically used to represent pixel values in the digital domain. A useful insight is that the number of bits required for the representation is not just for SNR, but also dynamic range. Wide dynamic range capabilities are often required when transmitting and processing physically-based signals. This is especially the case with visual signals, as discussed in Section 2.3.

Even though analog computations have limited SNR because of mismatch, certain circuit topologies can inherently process signals with a wide dynamic range. These analog approaches are well suited for imaging applications because the relative level of actual information in each individual pixel is usually not very high. The underlying information is usually contained in the collection of several pixels and dynamic range abilities represent the capacity to capture and to process that information. It therefore becomes natural to perform basic signal conditioning, classification, and data refinement in the analog domain before passing essential information to the digital domain for complex processing.

Early data reduction also has the advantage that it can reduce throughput requirements and power consumption in later parts of the communication and processing chain. When done in the analog domain, it reduces the requirements on digital-to-analog converters (DACs), which consume significant power. Again, this is critical in video applications because the data volume is large. Even the cost of interchip communication should be considered in video applications. This early data processing at the sensor interface is only usually achievable with analog circuits.

CHAPTER 2

CMOS IMAGERS

Modern CMOS imagers are opening a new field of possibilities for image sensing and processing. CCD imagers have largely dominated the imaging market and produce the highest-quality results, but they have the limitation of needing special processes that do not allow for high levels of on-chip integration. Also, CCDs require high voltage generation and consume more power than CMOS imagers. CMOS imaging technology, on the other hand, can be implemented on standard, relatively low-cost CMOS processes. This allows standard analog and digital circuitry to be integrated with the image sensor onto a single chip. This opens many opportunities for mixed-signal image processing and has already enabled circuit advancements that are making CMOS imagers a new standard in high-end consumer cameras. A system-on-a-chip approach offers the ability to perform complex algorithms in a small area with higher speed, lower power, and lower noise. Advancements in CMOS imaging will allow for new imager applications and paradigms of image processing. These low-cost smart imagers will facilitate the development of complete vision systems that can be integrated in low-power and mobile applications.

Functionally, when comparing CMOS imaging technology to CCD imaging technology, a distinctive advantage is the ability to randomly access the pixels. While CCD imagers use a fixed, sequential access scheme to read the image, CMOS imagers can take advantage of random access to pixel data. The design of a random access CMOS imager is discussed in [9].

2.1 Basic Photoreceptor Circuits

The basic CMOS photoreceptor is a reverse-biased PN junction. As commonly known, a reverse-biased diode normally conducts very little current. The reverse-biased voltage connected to the diode adds to the built-in potential to create a barrier for charge carriers.

With enough reverse-bias voltage, this barrier is large enough that very few carriers have enough energy to overcome and cross the barrier. This sets the conditions to allow light-induced currents to be predominant.

When photons strike near the junction, they can add energy to weakly-atom-bond electrons [10]. If enough energy is imparted on an electron by the light, the electron is freed from its bound state and can move freely. A primary factor for determining this is the wavelength of the light. The wavelength, along with the velocity of light, sets the frequency, ν . The frequency relates to the energy by a relationship described using Planck's constant: $E = h\nu$.

The freed electron and the vacancy it leaves behind are known as photogenerated electron-hole pairs. If the carriers are generated inside the space-charge region of a diode, the electric field there quickly pulls electrons and holes in the proper direction to create reverse-bias current. This photon-induced current flow is what is used for measuring the light intensity at the sensor. If the carriers are generated outside the space-charge region, they must randomly diffuse into the junction without first recombining with an opposite charge carrier. A larger space-charge region in the diode tends to successfully capture more light and convert it to current. The efficiency of capturing and converting light is measured as quantum efficiency. Factors affecting the space-charge region are the doping levels and the reverse-bias potential applied.

In the photodiode, the current flow is proportional to the number of photons that fall on or near the junction. This allows the photodiode to act as a light-controlled current source in a circuit. It is not a perfect current source, since the voltage across the junction affects the current flow somewhat. However, the effect is relatively minor. This behavior is also experienced when using a transistor as a current source, which has current flow controlled by not only the gate-to-source voltage, but also the drain-to-source voltage. This is called the Early effect. It is modeled in small-signal model as a resistor in parallel with to the current source. A similar resistance is used in a small-signal model for a photodiode [11]

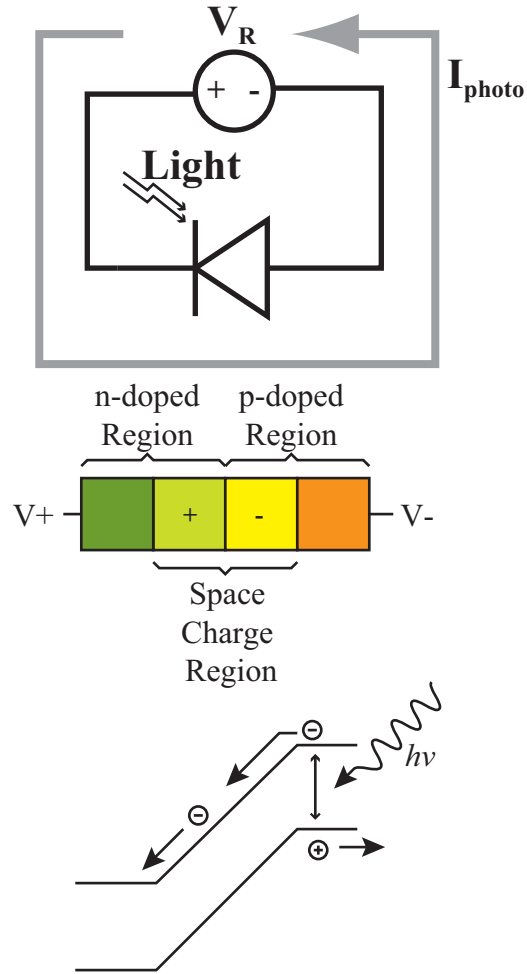


Figure 2.1. Photo diode

and a diode.

To use the current from the photodiode, a current amplification or I-V conversion must usually be performed. Figure 2.2 shows some basic photoreceptor circuits. Figure 2.2(a) shows the basic current flow, I_{photo} , which is proportional to the light falling on the reverse-biased PN junction. Figure 2.2(b) shows a photodiode used as a current source in a source follower configuration, producing a logarithmic light-to-voltage conversion. To understand the behavior of this configuration, one must realize that the current flowing through the photodiode in typical imaging applications is on the order of nanoamps or picoamps. This small current flow through an NFET mandates a subthreshold analysis of the circuit. In the

subthreshold region, the current flow through the transistor is described by Equation 2.1:

$$I_D = \frac{W}{L} I_t e^{-\frac{V_t}{U_t}} e^{\frac{\kappa V_g - V_s}{U_t}} e^{-\frac{\kappa V_g - V_d}{U_t}} \quad (2.1)$$

Since the source voltage appears in an exponential term in the current equation, the output of this circuit will change logarithmically with changes in current. Outputs that are logarithmically related to inputs have compressive characteristic that is often desirable. This compressive behavior means that the circuit can handle input changes over several orders of magnitude while keeping the output changes at reasonable levels. Imaging applications often encounter light levels that vary by several orders of magnitude, even in the same frame. Logarithmic compression can allow for successful capturing and processing of these widely varying light intensities. Figure 2.2(c) shows a more typical logarithmic conversion that uses a diode-connected PFET to give a voltage output that is logarithmically proportional to the light level.

The last circuit, illustrated in Figure 2.2(d), shows one of the most widely established CMOS imaging technologies, the Active Pixel Sensor (APS) [12]. This circuit takes advantage of the inherent capacitance of the PN junction to perform current integration for producing a voltage. To begin, the reset transistor resets the capacitor, leaving it in a charged state. Then, the reset transistor is turned off and the photodiode drains the capacitor at a rate proportional to the light level. The voltage on the capacitor is actively buffered through a source follower amplifier configuration. Here, the term active refers to the ability to generate an output signal with more power than is provided by the input signal. This power is drawn through a power input port connected to a power supply.

2.2 Active Pixel Sensor (APS) Imagers

Active pixels sensors are the most commonly used CMOS image sensor technology. To evaluate the basics of the technology, an APS pixel was fabricated and tested. The layout included the row select transistor needed for use in an array. Typically, the NFET bias

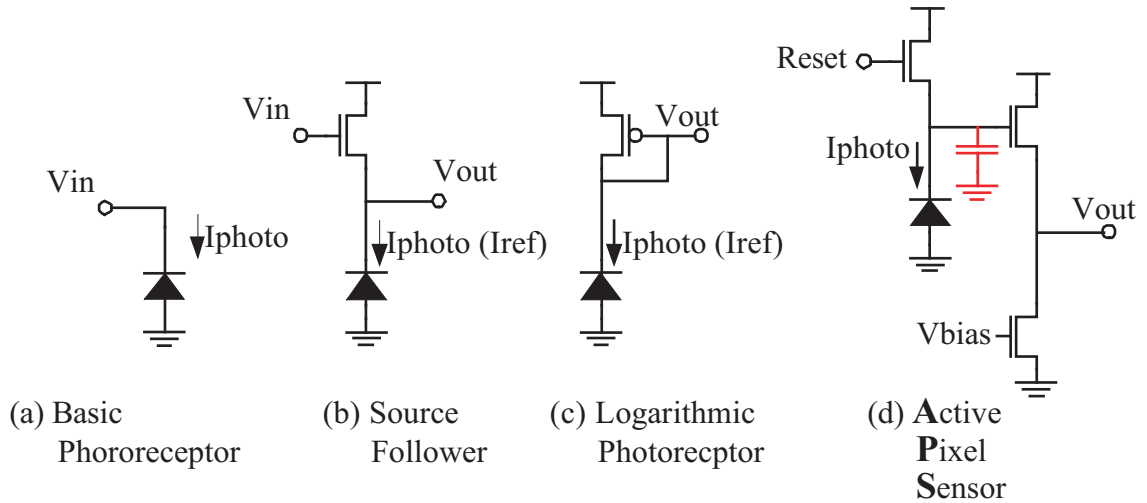


Figure 2.2. Basic photoreceptor circuits. (a) The basic photoreceptor is a reverse bias PN junction which conducts a current proportional to the amount of light falling on the junction. (b) The photoreceptor can be used as current source in configurations like the source follower and (c) the logarithmic photoreceptor, which both perform logarithmic compression in the current to voltage conversion. (d) The Active Pixel Sensor configuration uses an active amplifier to generate the output. In the APS circuit the current is integrated on an implicit capacitor and that voltage is given to the active amplifier.

transistor for the output amplifier is shared for a column of pixels, as illustrated in Figure 2.3.

Light filters were used to test the response of the pixel. Figure 2.4(a) shows the transient voltage of the APS. The initial jump in voltage occurs with the reset signal, as seen in the figure. When the reset signal is lowered, the combination of a capacitive coupling effect and a charge feed-through effect lowers the voltage on the diode capacitor. This is observed as a sudden, small drop on the output voltage. Following this drop is the expected integration of the current of the photodiode on the capacitor, causing the voltage to fall. Using a fixed light source and various light absorption filters, the light intensity on the sensor was varied to produce seven levels of light with total variations over two orders of magnitude. The brightest light level at the sensor occurred when using no filter, and is denoted by 100% transmission. The lowest light level was created using a light filter that passes 1% of light through. As expected, the integration slope is linearly proportional to the light intensity falling on the photosensor. The vertical, dotted lines in Figure 2.4(a) denote the region

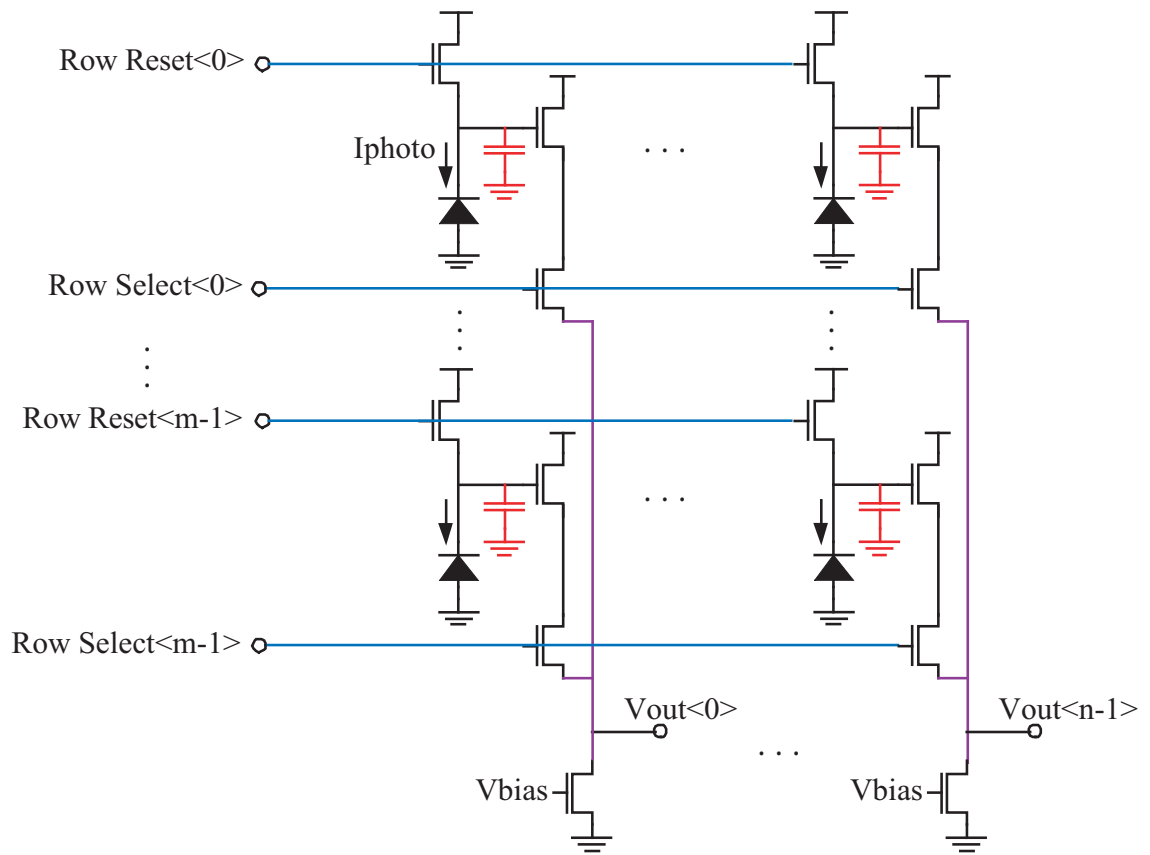


Figure 2.3. Active Pixel Sensor (APS) array

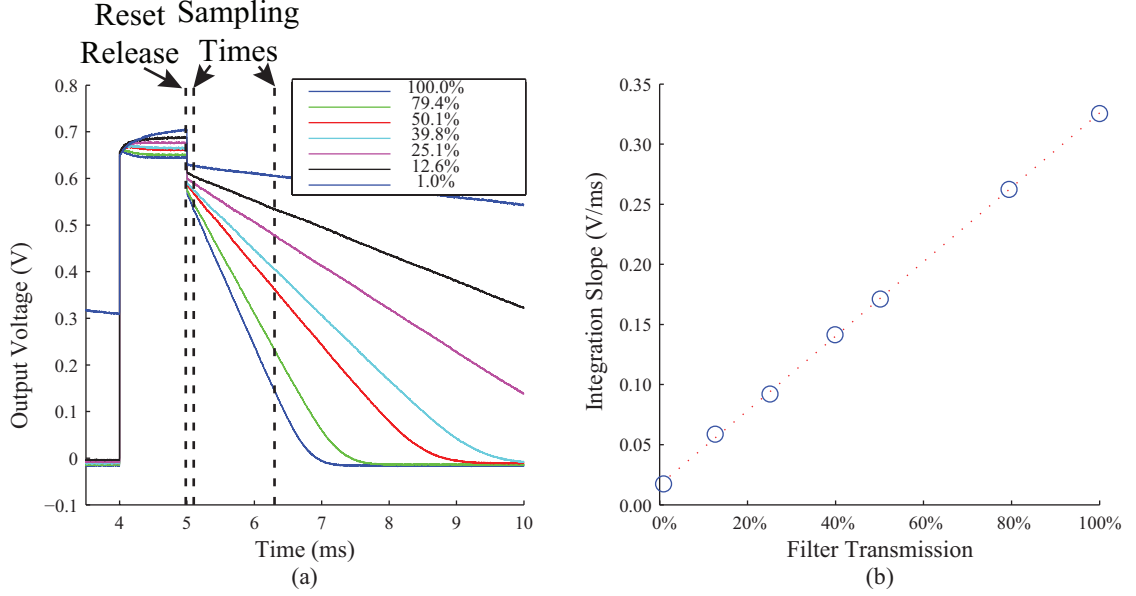


Figure 2.4. Measured APS pixel operation. (a) APS transient curves with varying light using light filters. (b) Extracted slopes

used for slope extraction and the expected the results are shown adjacently in 2.4(b).

2.3 High Dynamic Range Imaging Techniques

Wide dynamic range performance is critical in real-world sensors, especially in imagers. The point is made by Yadid-Pecht and Fossum [13] that illuminations range from 10^{-3} lux at night, to 10^2 or 10^3 lux indoors, and to 10^5 lux outdoors on a sunny day. Though global adjustments could be made depending on the interscene light conditions, there is the more difficult challenge of intrascene dynamic range. An imager viewing indoor and outdoor conditions in the same frame must be able to sense widely ranging signals on a short time scale. In conventional APS systems, long integration times undesirably allow voltage saturation, and short integration times cause the loss of dim light resolution. To address this problem, Yadid-Pecht and Fossum used dual readout chains to read the pixels twice in the same frame to achieve two different integration times.

Taking an alternative approach, Delbruck [14] uses focal-plane processing to solve the problem of intrascene variations. Local, per-pixel pixel circuitry was used to adaptively

scale input signals according to local dynamics. With each pixel adapting individually, many magnitudes of light could be sensed simultaneously. It was also shown that the circuitry overhead can be reduced by sharing the adaptive circuitry among groups of pixels, performing regional adaptation. .

Yang et al. also use focal-plane circuitry to solve the intrascene problem. They use a ramp-compare ADC [15] that is shared between four pixels in the focal plane, to convert signals. By varying the ramping cycles they can achieve many different conversion scales. Other techniques involve intrascene variance of integration time [16] by controlling when a reset occurs for a given region of pixels. This requires an intelligent controller with a memory.

2.4 Focal-Plane Processing

Neuromorphic VLSI is a field where circuits and systems are designed that mimic the behavior or structure of biological systems in some way. In the neuromorphic community, focal-plane processing became a focal point for an abundance of research. Focal-plane processing approaches move some processing traditionally done in DSP hardware to the architectural level of the pixel itself. This offers some unique computation and memory advantages by creating a distributed processing network.

A sample focal-plane processing approach to image edge enhancement is shown in [17], which uses several transistors in the pixel plane to emulate diffusion, mimicking biological synapses. Figure 2.5 shows two approaches for implementing an edge enhancement with a 3x3 kernel. In a traditional digital approach, computing the convolution at the center element requires that all nine data values be read and stored in memory. Later, the memory is accessed as calculations are performed to produce the final result. In the second approach, the convolutions are calculated in parallel at each pixel, while sensing, and the result is read directly from the pixel array. If more processing must be done, the neuromorphic scheme has efficiently segmented and organized the computations. This hierarchy of

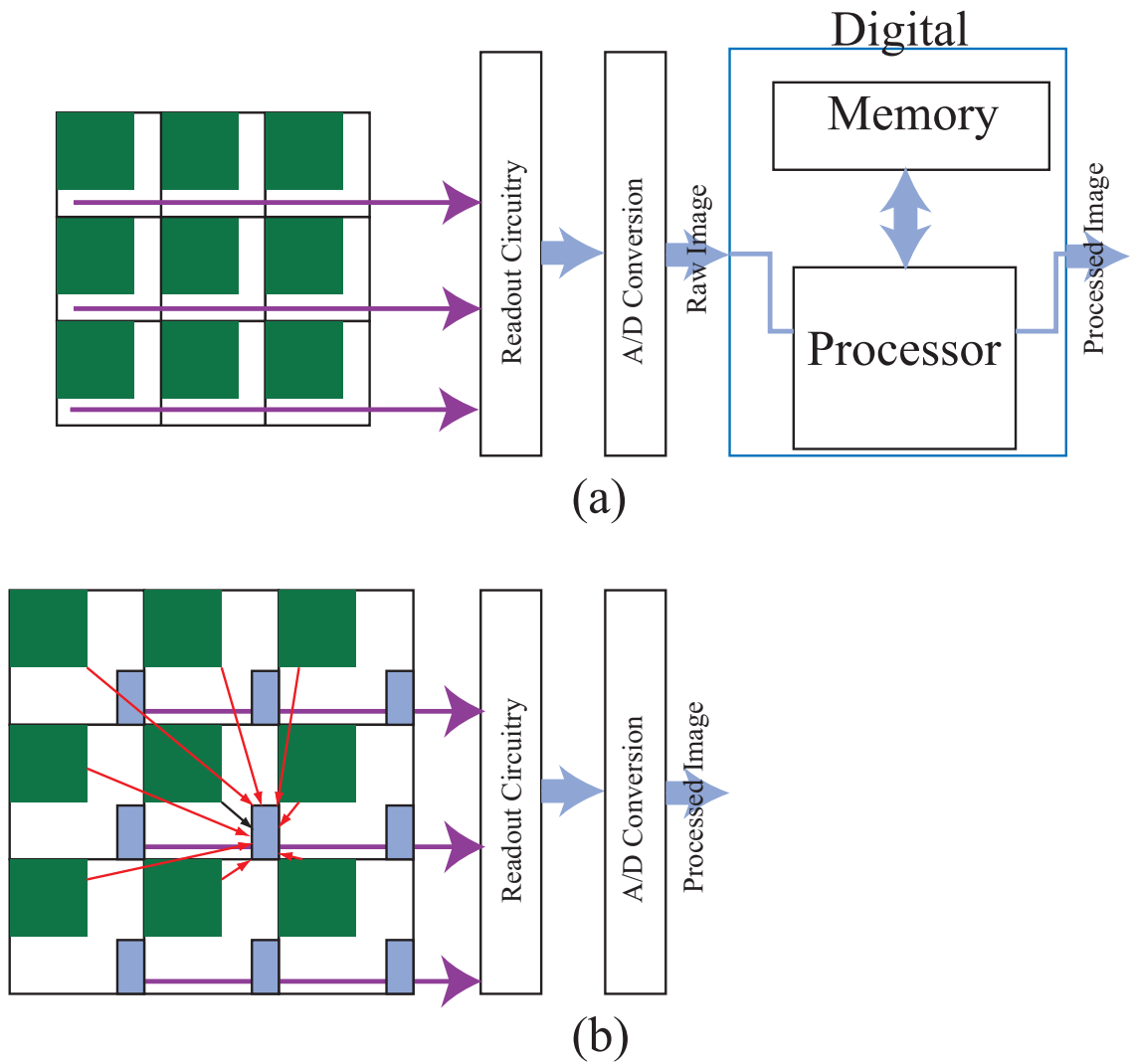


Figure 2.5. Architecture of traditional vs. focal plane processing. (a) In traditional imagers a certain percentage of area is photosensor and the rest is for control and readout. (b) In neuromorphic sensors, processing circuitry is added to the pixels, usually with intercommunication between pixels.

computation is similar to that seen in biology, especially in vision. Placing computational elements in the pixels comes at the cost of a reduced fill factor, which is the percentage of the pixel plane area that is used for the photosensors. The non-photosensor area of the pixel is sometimes referred to as a “dead” region [16] and causes loss of detail or aliasing. Some neuromorphic imagers have fill factors less than 5%, meaning that less than 5% of the physical pixel area is photosensitive.

The advantage of focal-plane processing is the elimination of the digital memory and processor, which typically consume more power for the same level of computation. Transmission power can also be saved. Chi et al. [18] use pixel-level ADCs with change detection. The created system converts or senses only changes in the image, and consequently transmits less information digitally.

2.5 Integrated Sensing and Processing and Intelligent ICs

As discussed later, there are significant advantages to reducing data bandwidth requirements early in the processing chain. There are also significant advantages inherent to performing several operations on a single IC, since interchip communication is expensive. While it is of primary importance to mobile and large distributed sensor networks, any system benefits from an efficient utilization of resources. Therefore, sensors designed to efficiently transduce only salient information and instead of raw data are desirable. As Figure 2.6 suggests, the goal ultimately is to sense information instead of data. Ultimately the sensors need to be intelligent enough to incorporate context. This context includes local observations, as well as the observations of other sensors, knowledge passed along from processing components, and any user-programmed knowledge.

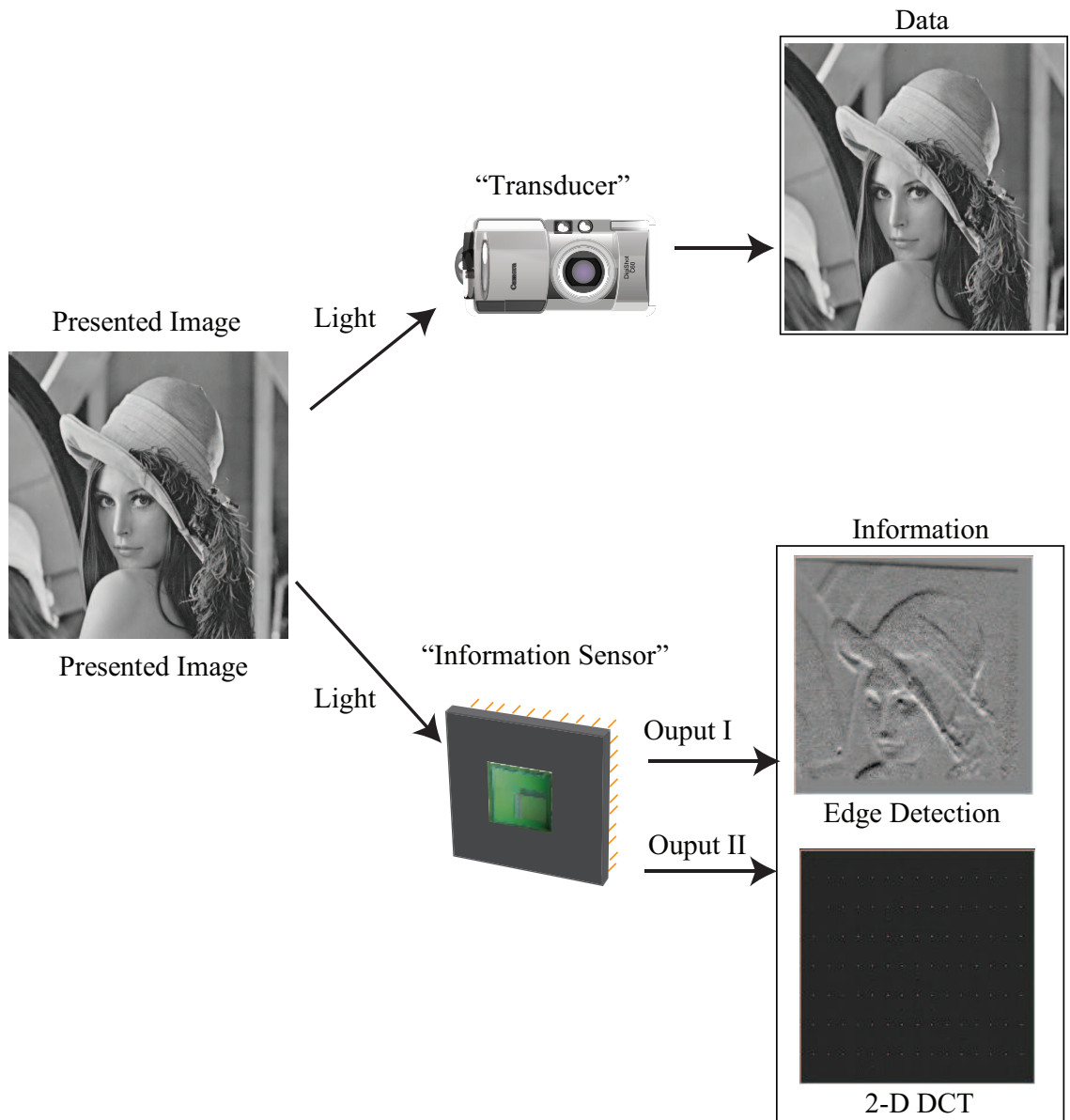


Figure 2.6. Information Sensor. The standard sensor is regarded and utilized as a transducer that converts a physical signal into an electrical signal with high fidelity to be processed later down the line. This work creates a sensor which conveys information instead of raw data. The ultimate goal is to create smart sensors with the the ability to selectively pass context-based information based on intellegent communication with other sensors and processing components.

CHAPTER 3

SUBTHRESHOLD CONDUCTION AND FLOATING-GATE TRANSISTORS

Developments in reprogrammable analog floating gate transistors have allowed compact, power-efficient implementations of analog memory and unique computational abilities that have enabled a variety of non-traditional analog circuit topologies [19]. Figure 3.1 depicts the basic structure of such a floating-gate transistor element. The figure shows a PFET transistor utilizing a floating gate. The gate of the transistor has no DC path, resistive or inductive, to ground. It only has conduction to other nodes through capacitive coupling. The node itself is a piece of polysilicon completely insulated by silicon dioxide. The unique computational abilities arise from a combination of the capacitive-coupling properties, programmability, and use of the FET in the subthreshold conduction regime.

3.1 Subthreshold Transistor Modeling

The following model for a MOS transistor operating in the subthreshold conductance regime is assumed for the discussions in this document:

$$I = \frac{W}{L} I_t e^{-\frac{V_t}{U_t}} e^{\frac{\kappa V_g - V_s}{U_t} - \frac{\kappa V_g - V_d}{U_t}} = I_0 \left[e^{\frac{\kappa V_g - V_s}{U_t}} - e^{\frac{\kappa V_g - V_d}{U_t}} \right] \quad (3.1)$$

The transistor's current in subthreshold operation is predominantly a consequence of diffusion, as opposed to drift, of carriers across the channel. The number of carriers available to diffuse is determined by a Fermi-distribution and a potential barrier controlled by the transistor node voltages. In many ways, the operation is more similar to that of a bipolar junction transistor (BJT) than that of an above-threshold MOSFET. As a consequence of these characteristics, there is an exponential relationship between current and voltage. An intuitive explanation of this behavior can be found in [20].

There are both forward and reverse components to the diffusion current. However, when

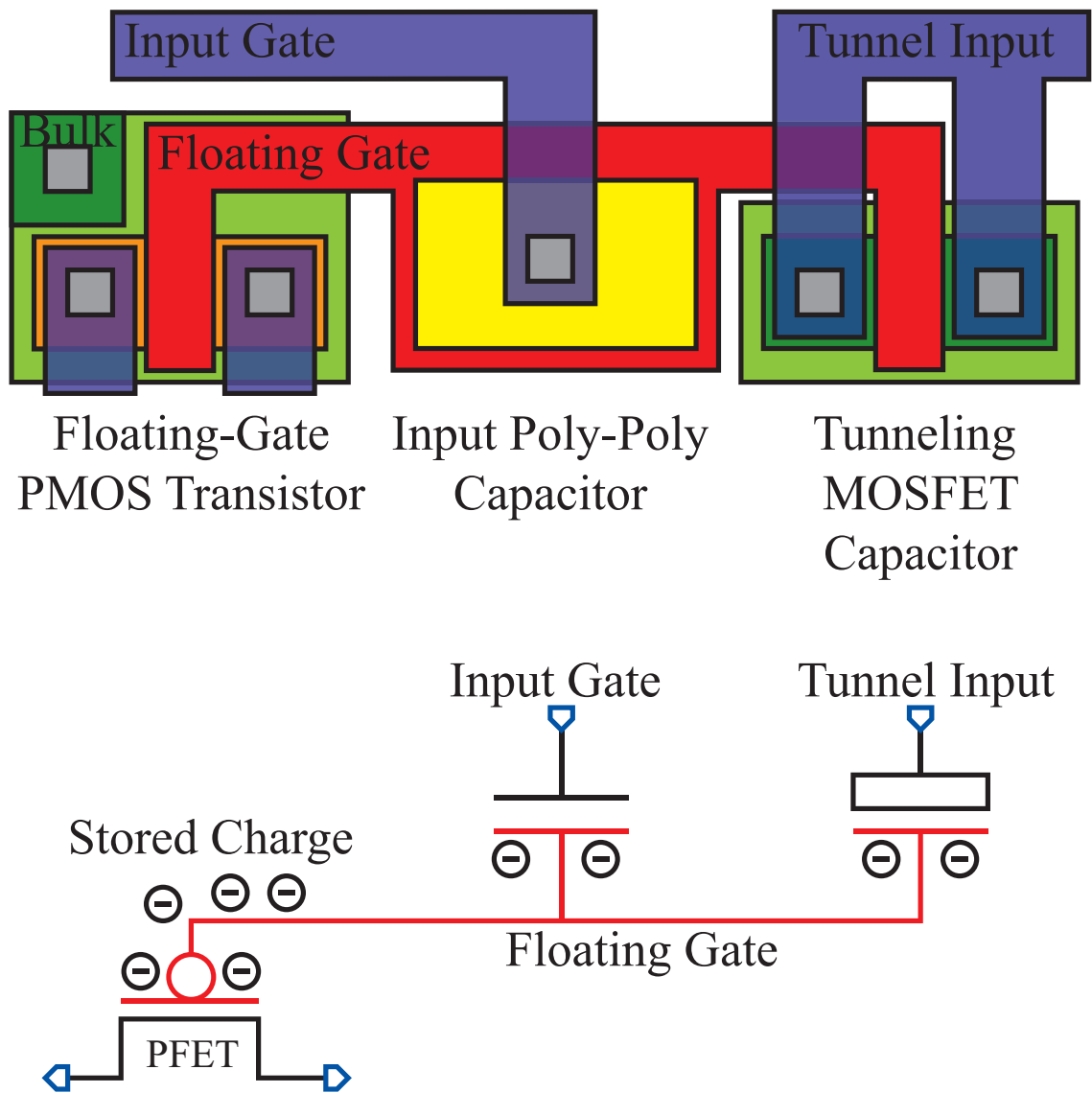


Figure 3.1. Reprogrammable floating-gate transistor

$V_d - V_s$ becomes large, the forward component dominates, and the transistor is considered to be in saturation. The second exponential term is dropped, and another term is added for channel length modulation (Early effect):

$$I = I_{n0} e^{\frac{\kappa V_g - V_s}{U_t}} e^{\frac{V_d}{V_A}} \quad (3.2)$$

A saturated PFET model is as follows:

$$I = I_{p0} e^{\frac{\kappa(V_{well} - V_g) - (V_{well} - V_s)}{U_t}} e^{\frac{(V_{well} - V_d)}{V_A}} \quad (3.3)$$

3.2 Subthreshold Floating-Gate Transistor Operation

A subthreshold-operating FGPFET is modeled by an equation similar to Equation 3.2, but instead V_g becomes a superposition of several components determined by capacitive coupling:

$$V_g = V_{gin} \frac{C_{in}}{C_{total}} + V_s \frac{C_{ov}}{C_{total}} + V_d \frac{C_{ov}}{C_{total}} + V_{offset} \quad (3.4)$$

V_{offset} represents a summation of several static terms such as the tunneling, substrate, and well voltages. Another term comprising V_{offset} is the quantity of charge stored on the floating node, Q , which adds a voltage $\frac{Q}{C_{total}}$. This term is usually constant, except when it is being explicitly modified through techniques discussed in the next section. Because the V_g term is in an exponential term in Equation 3.4, the floating-gate transistor conducts a current proportional to the exponential summation of several voltages and a programmable offset. The exponential summation allows for multiplicative operations and the programmable offset provides the critical ability to tune circuit behavior.

3.3 Reprogrammable Analog Floating-Gate Transistors

The reduction of charge stored on the floating gate is done using hot-electron injection. If a current flows through the transistor's channel while a large channel-to-drain potential

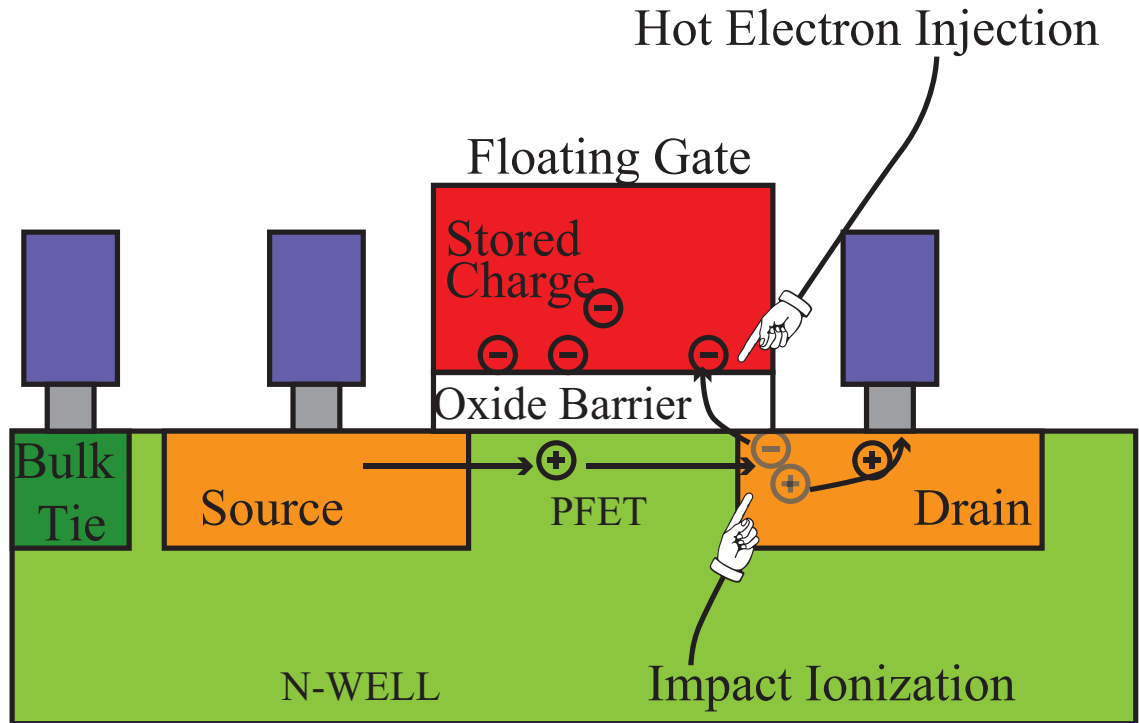


Figure 3.2. Hot-electron injection. Under the correct conditions, a carrier crossing the channel creates a hole-electron pair at the drain through a process called impact ionization. The electron created can have enough energy to overcome the thin oxide barrier and enter the floating gate, becoming a part of the stored charge.

exists, some carriers (holes) will fall into the drain with enough energy to create an electron-hole pair. The created electron may enter the channel and be pulled toward the gate. Then, if it has been imparted with enough energy, the electron can cross the thin oxide and enter the gate of the transistor, reducing the floating-gate charge.

Tunneling, on the other hand, increases charge on the node. Tunneling is performed by applying a large potential to the tunnel input node. This creates a field across the oxide at the tunneling junction large enough to induce tunneling of electrons through the silicon dioxide barrier to the tunnel input node.

Fortunately, many reprogrammable FGPFETs can be compactly tiled to create banks of analog memory and computational arrays. Figure 3.3 depicts a 2×2 array of FGPFETs. Assuming all the transistors have their sources connected to the power supply, there are

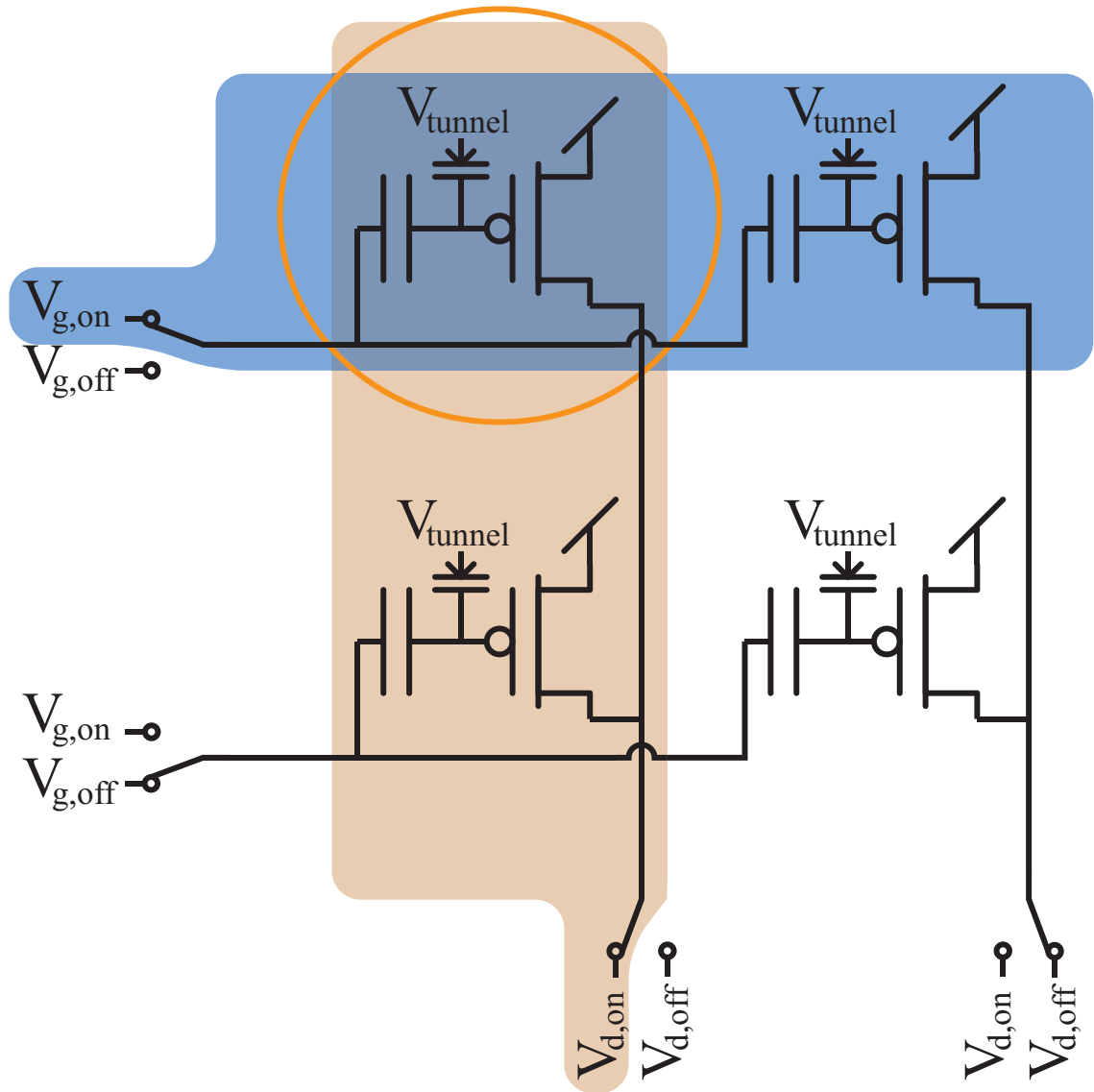


Figure 3.3. Floating-gate array programming. Programming a floating-gate transistor requires a sufficient gate voltage to turn the transistor on and a sufficient drain voltage to allow creation of hot electrons. These two controllable, necessary conditions allow a unique selection of a transistor in a two-dimensional array for programming.

two conditions that must be satisfied to program a transistor. The first condition for programming is that the gate voltage of the transistor must be such that the FGPFET conducts current. The second condition is that the drain must be at a low enough voltage to allow the creation of hot electrons. These two conditions allow the unique selection of a transistor in a 2-D array for programming.

Figure 3.3 shows a typical topology for a FGPFET array. Transistors share input gate connections along rows, which are multiplexed between an on-voltage and an off-voltage. Similarly, the drains are shared along columns and are multiplexed between two voltages. With this scheme [21], if only one row is given a gate input voltage that turns a row on, and only one column is given a low drain voltage, then only one transistor in the array will have both sufficient conditions for injection. Global erase is achieved using a tunnel voltage that is shared throughout the array. Typically, the programming cycle for an array involves a global tunnel followed by the injection of each transistor to individual levels.

CHAPTER 4

COMPUTATIONAL FOCAL PLANE

The core enabler of the computational image sensor presented here is a focal plane that can perform matrix operations on incoming images. The focal plane is composed of computational pixels that can sense light and perform computations. Each pixel performs light sensing, multiplication, and addition. The pixels in the array operate in parallel, under the control of periphery circuitry, to capture data and processes it using matrix multiplications. In this chapter, applications of matrix multiplication for image processing are explained, along with a theoretical and experimental study of the computational pixels that implement it.

4.1 Image Processing Using Matrix Operations

The separable transform image sensor can perform separable 2-D filtering on an incoming image. The capability can be described mathematically by the following matrix multiplications:

$$Y = A^T P B \quad (4.1)$$

Here, the matrix A^T defines how the columns of P are filtered or transformed and B defines the same for the rows. This formula can produce different operations, such as low-pass filtering, edge detection, and the Discrete Cosine Transform (DCT) (the fundamental operation of JPEG compression) [22]. The available parameters and the flexibility in the control of the application of computation allow this one IC to be programmed to perform a versatile set of operations, Figure 4.1. Some explanation of filtering and transformations using matrix operations follows in this chapter.

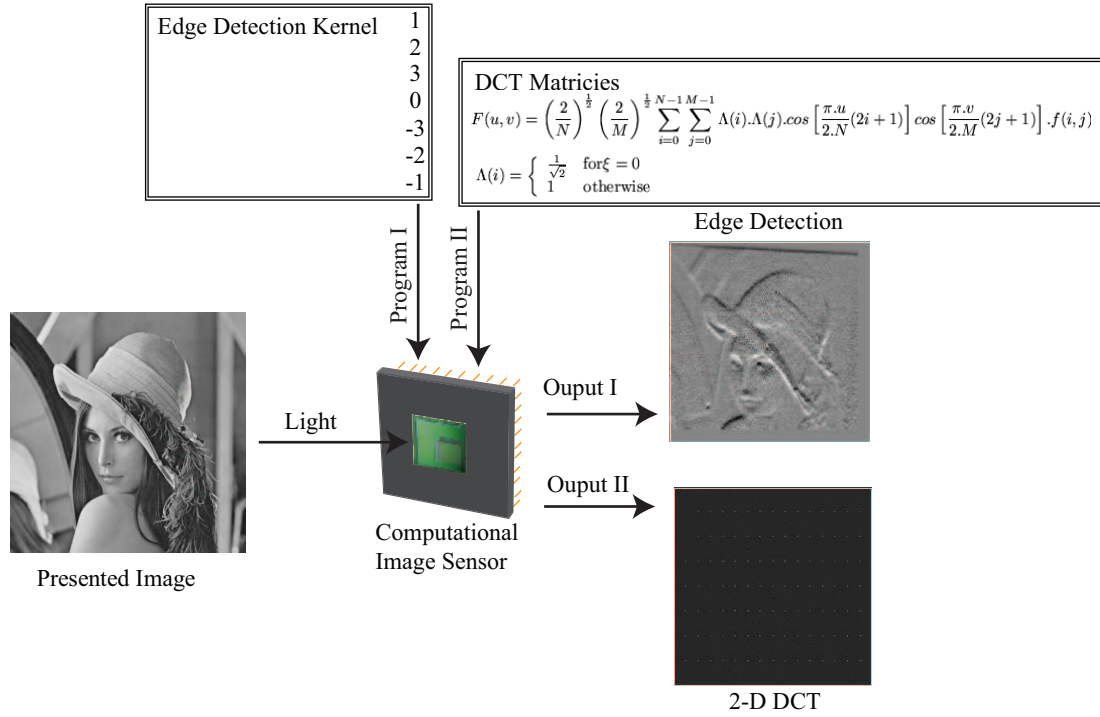


Figure 4.1. Reprogrammable computational image sensor. The imager has a fundamental computational capability to perform a matrix operation, $Y = A^T P_{\sigma} B$, on selectable subregions of the image, P_{σ} . By reprogramming the parameters and application of the operator, a verity of operations like edge detection and 2-D discrete cosine transformations are accomplished.

4.1.1 Signal Processing with Matrix Operations

A signal, which is simply a series of values, can be represented by a vector \vec{v} . If \vec{v} is a column vector, a matrix-vector multiplication, $A\vec{v}$, can represent a number of processing operations on the signal. Of particular interest are two operations: change-of-basis and convolution.

A change-of-basis (or change-of-coordinates) matrix is created by placing the new normalized basis vectors, written with respect to the initial coordinate system, in the columns of A and performing the matrix-vector multiplication $A^T\vec{v}$. A detailed description of changing basis is given in any basic linear algebra textbook. A motivation to perform a basis or coordinate change is that some coordinate systems allow certain calculations and analyses to be simplified. This is typically achieved by reducing the number of dimensions that must be operated on or considered. Aside from simplification of calculations and analysis, a more reduced representation of a signal is useful storage purposes. It is often the case that a coordinate system can be chosen such that certain dimensions of the data are commonly insignificant and can be discarded or stored with reduced resolution. This is the essence of data compression. The discrete-cosine-transform is one such commonly used coordinate transformation and is used in image processing for JPEG compression. In the process of JPEG compression, a change of coordinate systems is performed and then dimensions of the data are discarded or stored with reduced resolution if they do not significantly or desirably help describe the signal.

The second mentioned operation, convolution, is the foundation of most digital filtering: $y[n] = \sum_{k=-\infty}^{\infty} h[k]v[n-k]$. A convolution operation can be represented in matrix form by creating a convolution matrix, A , which has shifted versions of a convolution kernel in the rows of A . The convolution is then written as $A\vec{v}$. For instance, a convolution kernel of length three, could filter a signal \vec{v} of length eight by using one of the following matrices:

$$\begin{bmatrix} h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_2 & h_1 & h_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_2 & h_1 & h_0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_2 & h_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_2 \end{bmatrix}, \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 & 0 & h_2 & h_1 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & h_2 \\ h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_2 & h_1 & h_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & h_2 & h_1 & h_0 \end{bmatrix}$$

The first matrix creates a result that is of more length than the input, since the convolution spreads the information in \vec{v} . To preserve the vector length, typically either the last rows are left off or the second matrix is used. The second matrix is a circular convolution, which is based on an assumption that the signal is repetitive.

4.1.2 Two-Dimensional Image Processing with Separable Transforms

A two-dimensional discrete convolution is defined as:

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} P[k_1, k_2] h([n_1 - k_1, n_2 - k_2]) \quad (4.2)$$

If the convolution kernel h can be written as $h[n_1, n_2] = h_1[n_1]h_2[n_2]$, then the transform is separable. This means that the 2-D transform can be written and performed as two 1-D transforms:

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{+\infty} \left(\sum_{k_2=-\infty}^{+\infty} P[k_1, k_2] h_2[n_2 - k_2] \right) h_1[n_1 - k_1] \quad (4.3)$$

The convolution kernel is an outer product $h = h_1 h_2^T$:

$$\vec{h}_1 \vec{h}_2^T = \begin{bmatrix} h_1[0]h_2[0] & h_1[0]h_2[1] & h_1[0]h_2[2] & h_1[0]h_2[3] \\ h_1[1]h_2[0] & h_1[1]h_2[1] & h_1[1]h_2[2] & h_1[1]h_2[3] \\ h_1[2]h_2[0] & h_1[2]h_2[1] & h_1[2]h_2[2] & h_1[2]h_2[3] \\ h_1[3]h_2[0] & h_1[3]h_2[1] & h_1[3]h_2[2] & h_1[3]h_2[3] \end{bmatrix}$$

The condition of separability places this restriction on the structure of the convolution kernel, disallowing an arbitrary selection of 2-D kernel coefficients. In the case of changing basis, the requirement that a transform be separable creates the constraint that the change of basis be implementable as a series of independent, 1-D change of basis operations in the x and y directions. However, with the restriction of separable transforms, we are still left with a large set of operations and flexibility.

Just as in the 1-D case, matrix notation can be used to represent separable, 2-D transforms. Working with a 2-D signal, P , instead of a 1-D signal \vec{v} , we can describe an operation on columns of P in matrix notation:

$$Y = AV \tag{4.4}$$

An operation on both the rows and the columns is represented like this:

$$Y = A^T P B \tag{4.5}$$

The matrix A^T defines how the columns of P are filtered or transformed and B defines the same for the rows. The convention of using a transposed matrix, A^T , allows the same matrix to be used for A and B if the operation on rows and columns is to be the same.

4.2 Computational Pixel Operation and Characterization

A computational pixel element is shown in Figure 4.2 that uses a differential pair to perform a multiplication. The inputs are light and voltage, and the output is current. To analyze it,

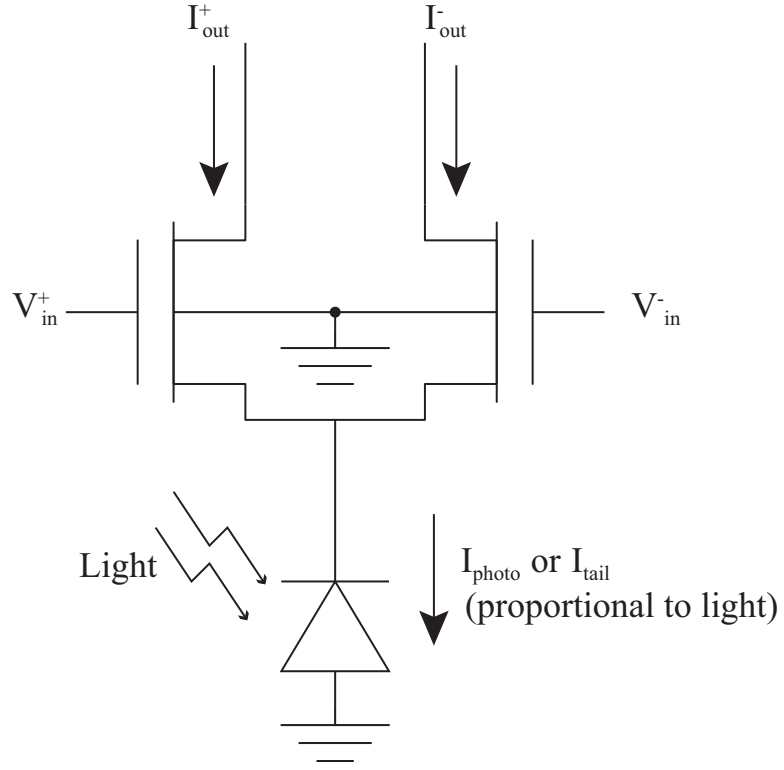


Figure 4.2. Differential pixel

one starts with the subthreshold, differential pair that behaves according to a hyperbolic tangent function as follows:

$$I_{diff} = I^+ - I^- = I_{tail} \tanh\left(\frac{\kappa(V_1 - V_2)}{2U_t}\right) \quad (4.6)$$

A brief set of characteristics of the curve created by a tanh function is as follows:

1. Crosses through the origin.
2. Behaves like a linear function near zero.
3. Levels out to constants -1 and 1 at the respective ends.

Replacing the tail current for the differential pair with a photodiode current, I_{photo} , and linearizing the tanh expression gives the following:

$$I_{diff} = I^+ - I^- = I_{photo} \left(\frac{\kappa(V_1 - V_2)}{2U_t} \right) = I_{photo} * M * (V_1 - V_2) \quad (4.7)$$

where M is simply the constant

$$M = \frac{\kappa}{2U_t}. \quad (4.8)$$

To definitively assure this expected behavior, several pixels have been characterized, both as single elements and as part of arrays. Figure 4.3 shows a single I-V sweep of a differential pixel in an array. The first thing to note is that the data curve in Figure 4.3 is not centered vertically at zero, but is instead offset to an extracted point $I_{mid} = \frac{I_{max} + I_{min}}{2}$. This offset is caused by a combination of factors, including parasitic currents and the effects of other pixels on the same readout line. The pixel's voltage offset, V_{offset} , is defined as the voltage where the pixel outputs the differential current I_{mid} . Ideally, this would be 0 volts. The current offset, I_{offset} , is defined as the differential current output minus I_{mid} , when the differential voltage input is 0 V. The linear range is the input voltage range in which the pixel's output current moves linearly with the voltage. The gain, G_m , is the differential voltage to differential current transconductance extracted from the slope of a line-fit in the linear region of the data. The photocurrent, or tail current, is extracted from the maximum and minimum currents, $I_{photo} = \frac{I_{max} - I_{min}}{2}$.

4.3 Validation of Voltage-Light Multiplication

The multiplication operation of the pixel is one between light intensity and a differential voltage, with a constant scalar multiplier $\frac{\kappa}{2U_t}$. This operation assumes that the current through the photodiode, and thus the height of the resulting tanh curve, indeed scales linearly with light. It is also expected that the slope in the linear region does the same. The concern would be that because the slope is affected by other parameters, namely, kappa (κ), it may not maintain its linear relationship to voltage and light. Figure 4.4(a) shows several I-V sweeps done at varying light intensities. The light intensity was controlled using light

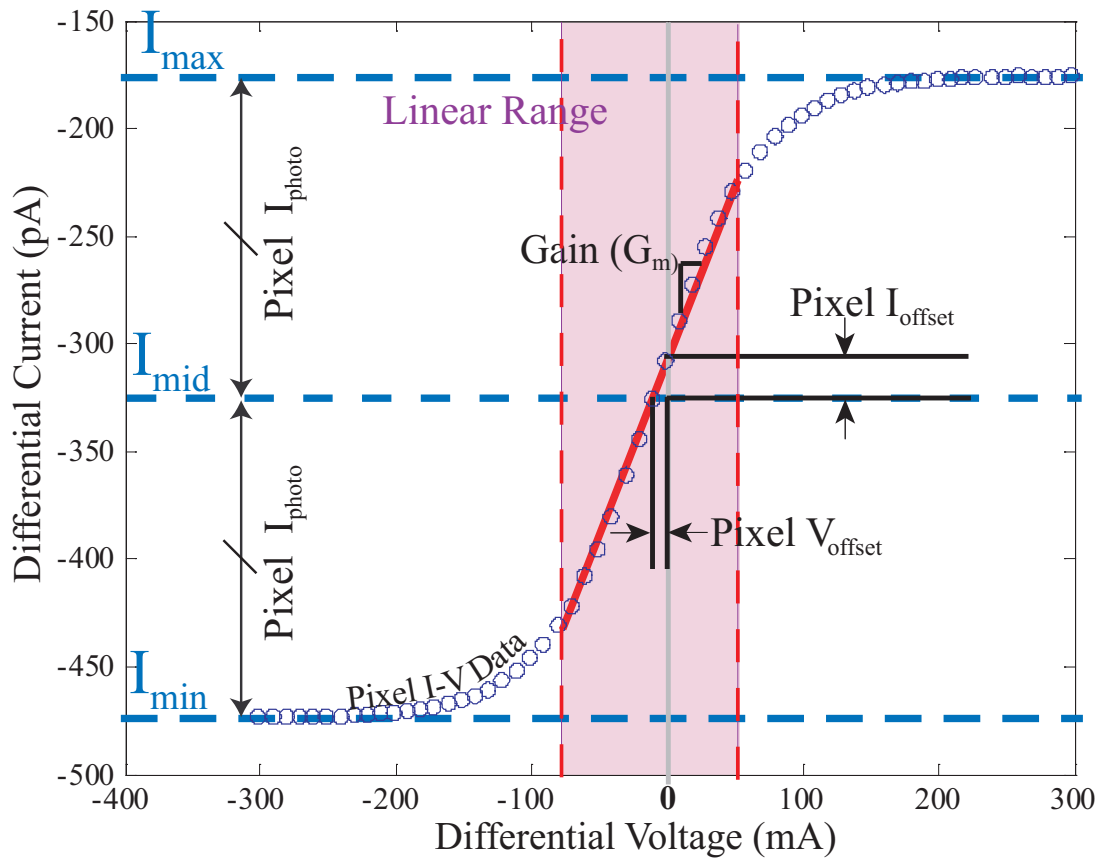


Figure 4.3. Pixel characterization. This is the typical I-V response and extracted parameters from a voltage sweep of a pixel located in an array.

absorption filters with known transmission levels. Transmission is meant here to be the percentage of light passing through the filter. The lowest light level was produced using a transmission level of 1%, while the highest level, 100%, was obtained using no filter at all. Therefore, the range of light intensities was varied two orders of magnitude. Since the pixel was in an array, it had associated current offsets that also moved with light intensity. Figure 4.4(b) shows the same curves with their offsets removed. Again, the offset is taken to be the average of the currents at the two extremities of the curves. To isolate the effect of the constant multiplier, $\frac{\kappa}{2U_t}$, the height of the curves was normalized and the results are shown in Figure 4.4(c). Smaller or larger values for κ would have caused corresponding changes in the slopes.

To validate the linearity of the output with respect to light, Figure 4.5(a) shows the tail current extracted from the height of the curves as a function of light intensity. The linear relation holds as expected. The offsets of the curves in Figure 4.4(a) are plotted in Figure 4.5(b). The linear relationship of the offsets results because the expected sources of the error, parasitic junctions and other pixels in the column, produce currents proportional to the light intensity. Figure 4.6 shows how the slope of the linear region scales appropriately with light intensity. These results help validate the proper multiplication operation of the pixel.

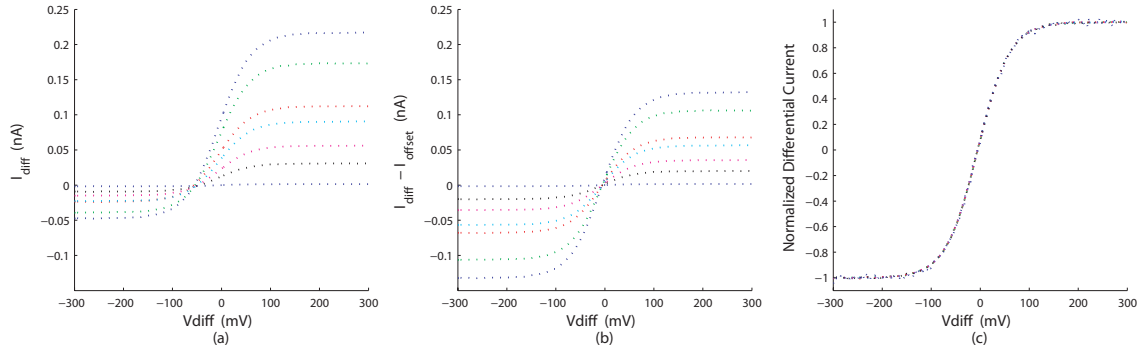


Figure 4.4. Pixel currents with varying intensity. These plots show output current vs. differential input voltage for seven light intensities that vary by up to a factor of 100 from the lowest to highest intensity using light absorption filters. (a) shows the original data; (b) shows the same curves with their offsets independently removed; (c) shows the same seven curves normalized. The last plot shows the consistency of the shape under varying light intensities. This verifies that the slope in the center scales with the height of the curve and that κ stays constant.

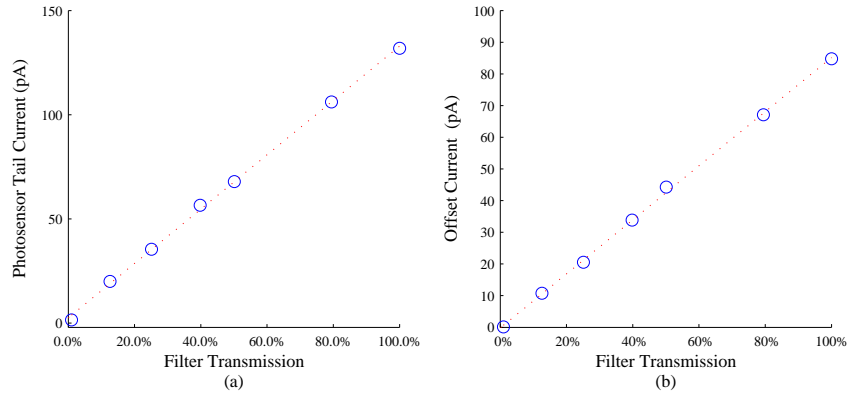


Figure 4.5. Photosensor tail current as a function of light intensity controlled using light absorption filters. (a) shows that the photosensor current feeding the differential pair is linearly proportional to the light intensity. (b) shows that the offset of the curve is also linearly proportional.

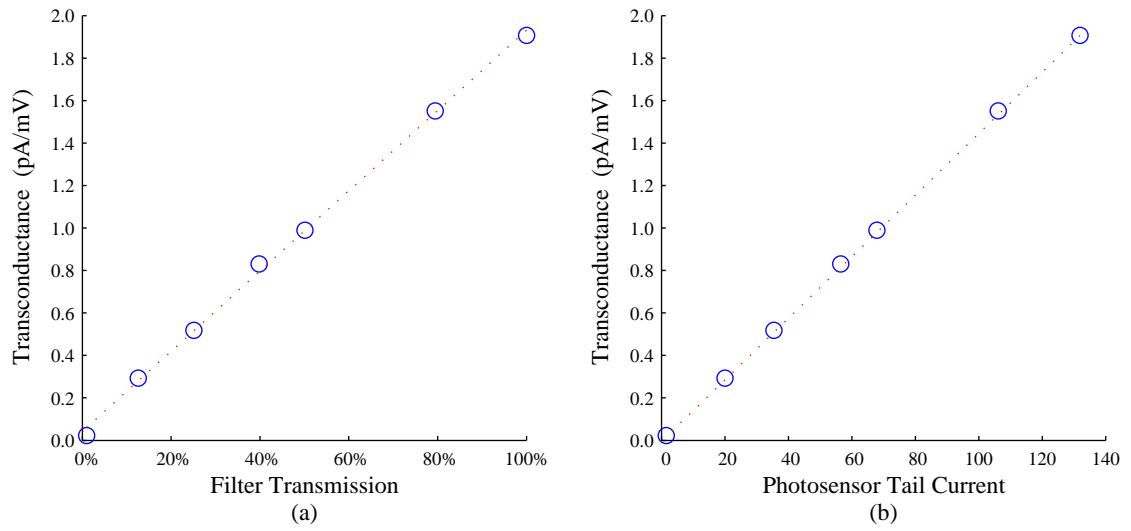


Figure 4.6. The transconductance of the differential amplifier related to light intensity and saturation current.

CHAPTER 5

COMPUTATIONAL SENSING SYSTEM ARCHITECTURE

In this work, a versatile computational imager with a core capability of performing separable transforms has been designed. Its capabilities include random access to the pixel plane, random access to stored transforms, and a flexible control of how the transforms are applied to different regions of the image. This enables dynamic and multiresolution field-of-view capabilities such as that found those in [23]. The system as shown in Figure 5.1 is entirely integrated on-chip, Figure 5.2, and is a progression toward larger resolution imagers. The current imager was implemented on a 22.75 mm^2 die in a standard $.35 \mu\text{m}$ CMOS process. The resolution is 256×256 with a pixel size of $8 \mu\text{m} \times 8 \mu\text{m}$. The system is composed of the following: a random access analog memory, row and column selection controls, a computational pixel array, logarithmic I-Vs, an analog vector matrix multiplier, and a bidirectional I-V converter. This work follows [6], which implemented a smaller, block-transform imager system. Each redesigned piece focuses on higher bandwidth and accuracy.

The fundamental capability of this imager can be described as a matrix transform: $Y_\sigma = A^T P_\sigma B$, where A and B are transformation matrices, Y is the output, P is the image, and the subscript σ denotes the selected subregion of the image under transform, Figure 5.3. The region σ is a 16×16 pixel block starting at an offset $(8m, 8n)$, where m and n are positive integers. Offsets smaller than the support region allow transforms that can reduce or eliminate blocking artifacts. For instance, separable convolutions with kernels up to size 8×8 can be used without suffering from artifacts.

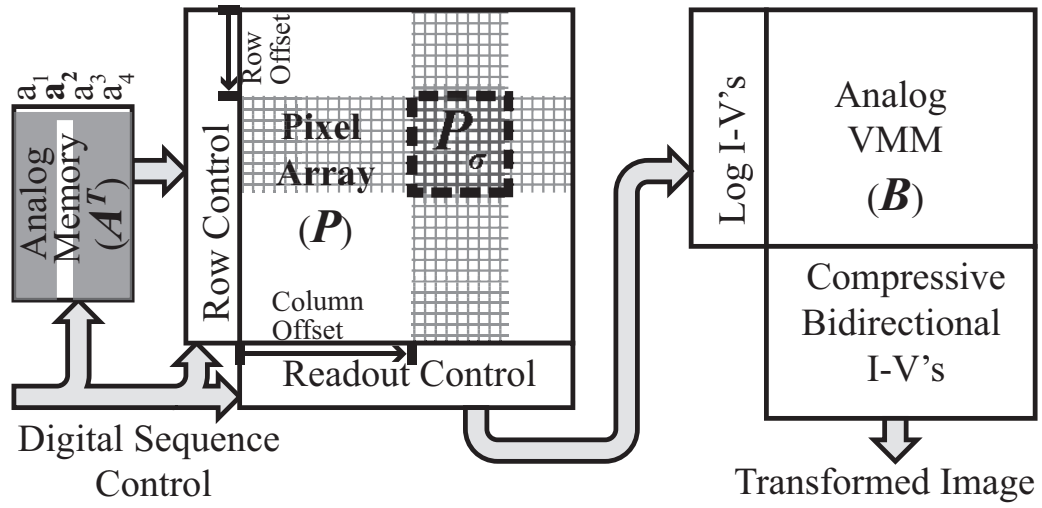


Figure 5.1. Computational imager sensor system level diagram showing the blocks of circuitry that implements the reprogrammable transform.

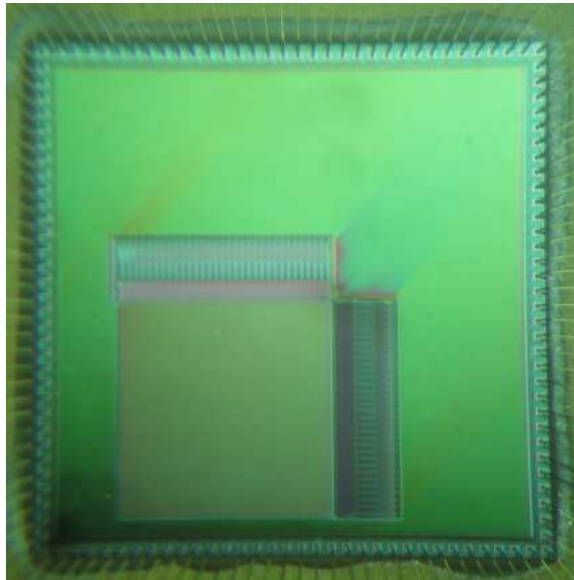


Figure 5.2. Die photograph of 256x256 imager.

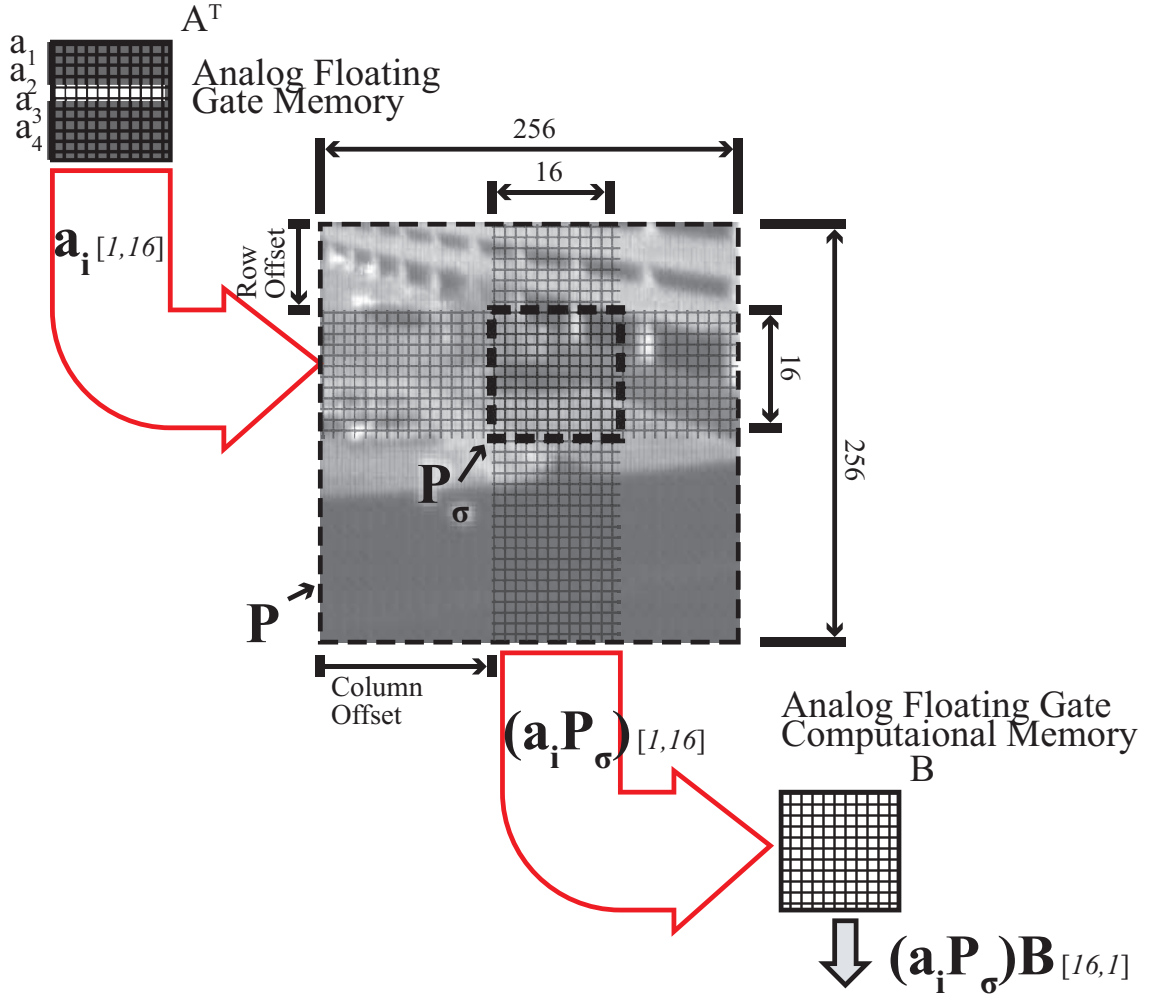


Figure 5.3. Computational imager sensor separable transform operation. The imager front-end is a reprogrammable random access analog memory. A selected row of coefficients, a_i of size 1x16, is applied to a corresponding set of 16 rows starting at an offset $8m$ where m is an arbitrary positive integer. Along those rows, each pixel senses light and converts it to a differential current with a multiplication factor determined by its row's coefficient. Along every row, currents are summed. A set of 16 column summations are selected, again with an offset of multiplicity 8, for multiplication by matrix \mathbf{B} . Thus, a vector $(\mathbf{a}_i P_\sigma) \mathbf{B}^T$ is computed where P_σ is the 16x16 sub-image undergoing transformation.

5.1 Computational Pixel Tile for In-Pixel A-Matrix Multiplication

Figure 5.4 shows a schematic of an 8×1 pixel tile. Each pixel is a photosensor and a differential transistor pair, providing both a sensing capability and a multiplication. Pixels along the same row of the imager share a single differential voltage input, which sets the multiplication factor for the row. Pixels along a column combine their output currents, producing a summation behavior. The tile also includes switches that group the 8 pixel rows to a common digital enable line. When disabled, the pixels are switched off of the column's output line and onto a separate line with a fixed voltage, thus reducing the output line capacitances and parasitic currents.

5.2 Random Access Analog Memory for the A-Matrix

A compact analog memory structure was used to implement the storage for the A matrix, Fig. 5.5. It uses analog floating gates to store the coefficients of the transform matrix, which means that no digital memory or DACs are required to feed the analog weighting coefficients to the computational pixel array. The use of several DACs along with digital memory would be costly in size and power. Building the memory storage element into the voltage generation structure avoids unnecessary signal handling and conversion, saving size and power.

The basic structure of the analog memory is an amplifier connected as a follower, Figure 5.5(a). However, one of the differential pair transistors has been replaced with a re-programmable bank of selectable analog floating-gate PFETs (FGPFET), Figure 5.5(b). Each FGPFET shares the same input, V_{bias} , but is programmed to a particular voltage offset that sets a desired output voltage. The programming procedure inherently avoids issues of voltage offsets due to mismatches in the transistors and in the op amp itself by directly monitoring the output voltage in the programming cycle instead of the floating gate voltage. [6] discusses the general use of FGPFETs. Here, generating 16 differential outputs requires 32 amplifier structures. The storage of a 16×16 differential value matrix requires

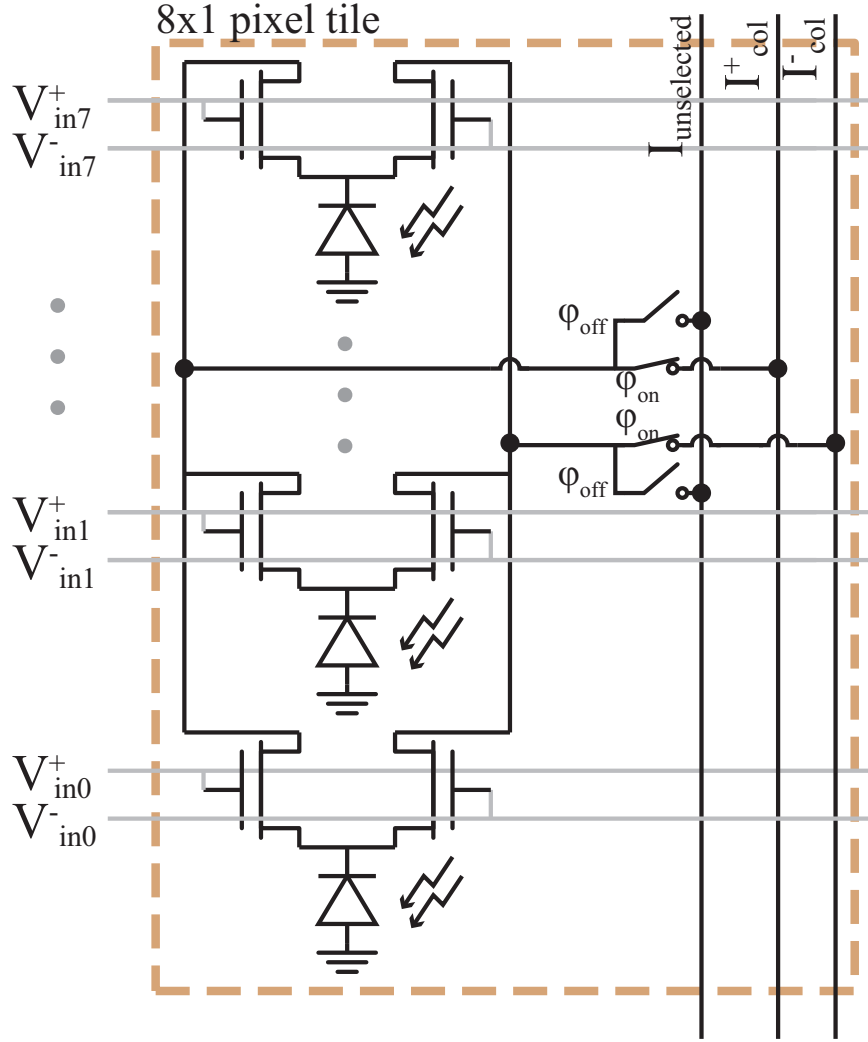


Figure 5.4. Pixel tile. Each tile contains 8 computational photo sensors and a set of switches which connect the currents to the column output or a separate fixed bias.

a total of 32 rows and 16 columns of floating gates. Stacking the amplifiers together creates a 2-D array of floating-gates in a convenient structure for parallel addressing and fits well into floating-gate array programming schemes.

5.3 Current Sensing and Processing for B-Matrix Multiplication

The back-end circuitry of the imager was designed to handle the large line capacitances and high dynamic range signals of the pixel array. Figure 5.5(b) shows logarithmic transimpedance amplifiers on the left that sense and logarithmically convert the pixel currents

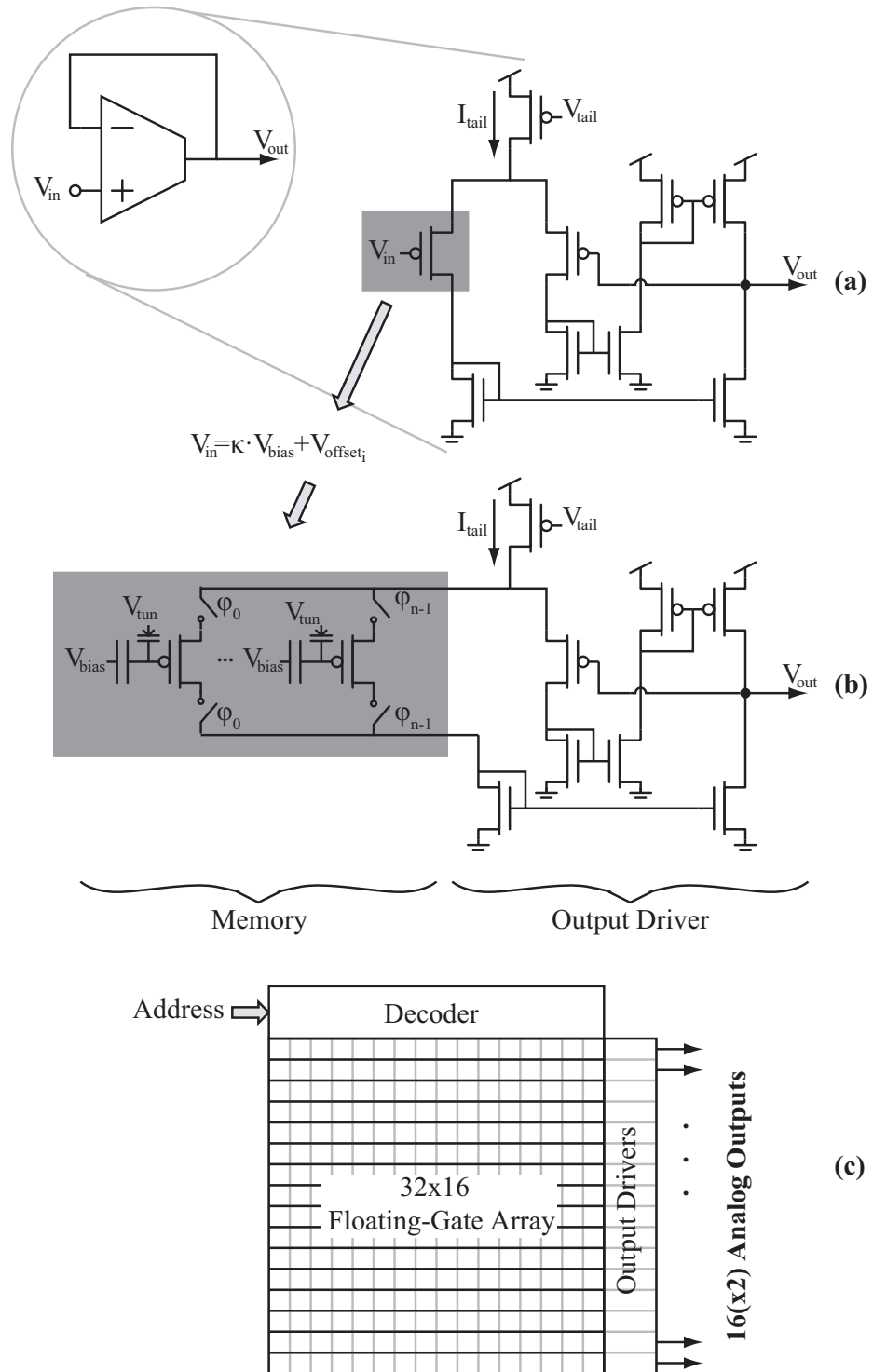


Figure 5.5. Random access analog float-gate biased memory. (a) Basic voltage buffer. (b) Input transistor replaced by selectable analog float-gate transistors. (c) Full analog memory bank.

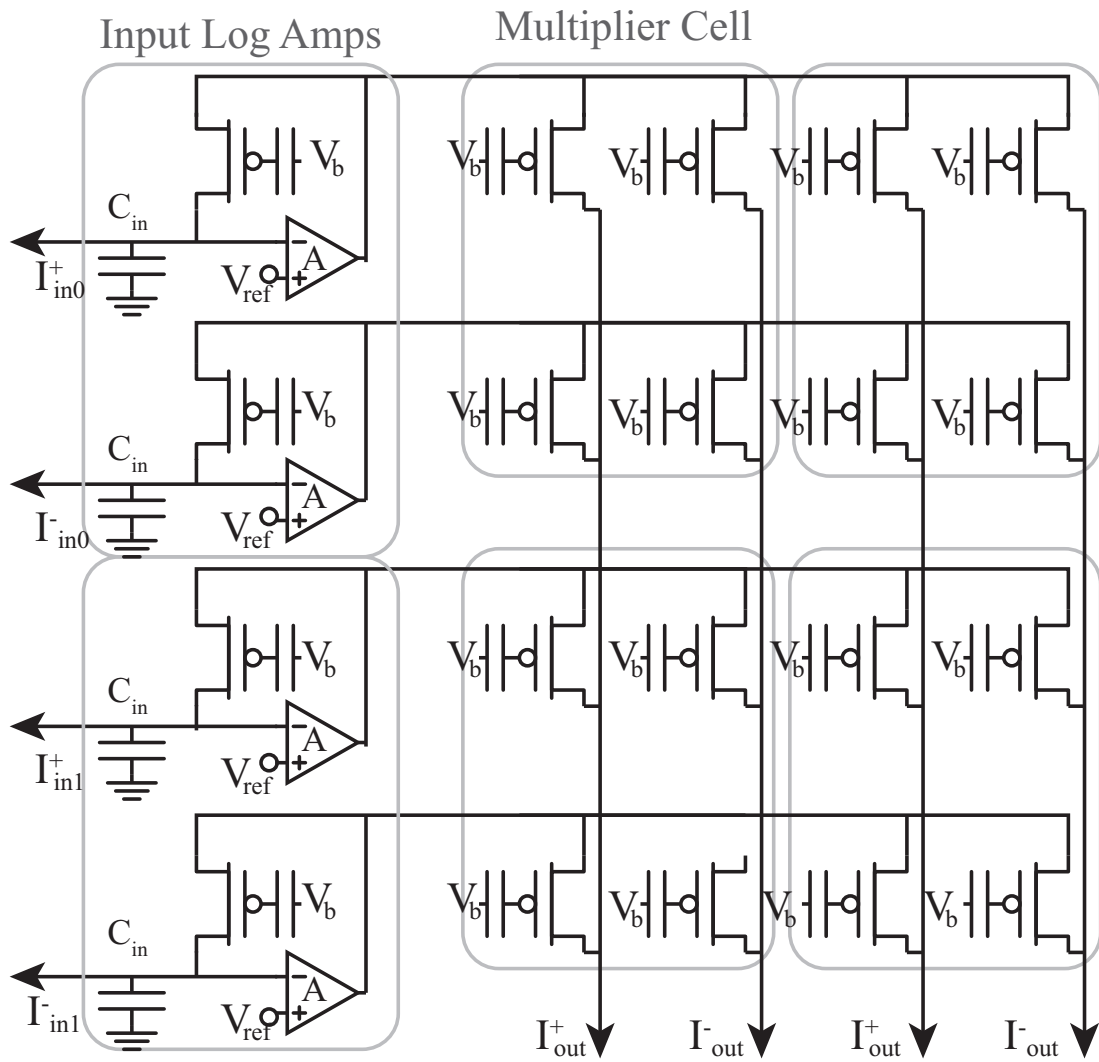


Figure 5.6. Fully differential 16x16 vector matrix multiplier (2x2 depiction here).

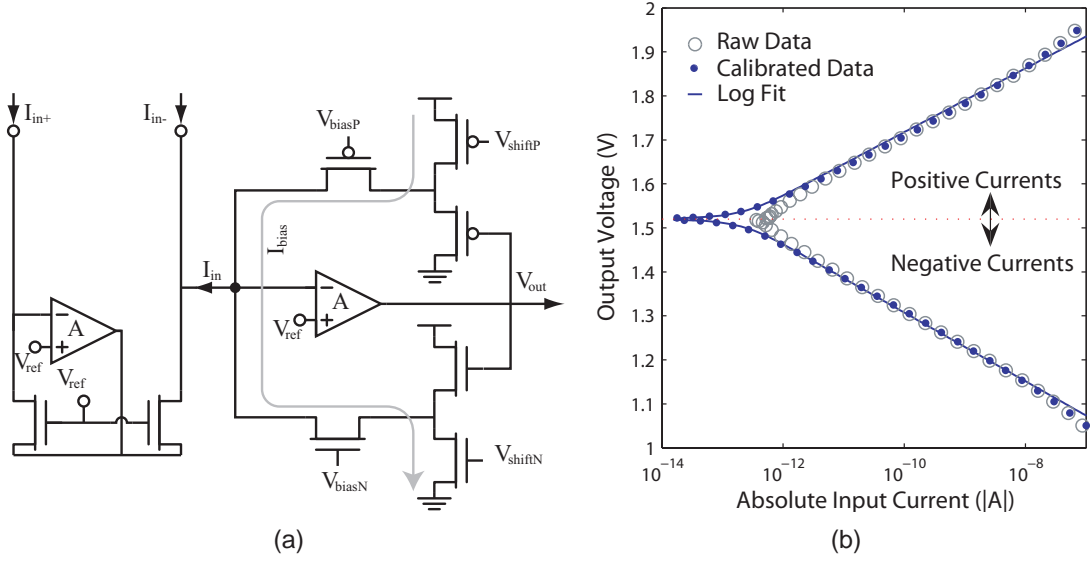


Figure 5.7. Differential to single-ended I-V converter. (a)Schematic (b) I-V conversion DC characteristic

to voltages. The logarithm is made possible by the subthreshold, exponential voltage-to-current relationship of the feedback MOSFET, much like a BJT or diode implementation[24]. The internal amplifiers, with labeled gain A , both buffer the outputs of the converter, providing the current for the load transistors, and create a large loop gain, fixing (clamping) the input voltage. The amplifiers lower the effective input impedance seen at the drain of the feedback transistor from $1/g_s$, where g_s is the subthreshold source conductance of the FGPFET, to $1/A g_s$. This low impedance is critical to sensing low currents in the presence of large capacitance. Also, the transfer characteristics of the transimpedance amplifiers can be matched by programming the FGPFETs. To greatly reduce power consumption, an automatic gain control (AGC) amplifier was integrated into the design that maintains speed and stability at various current levels. Because subthreshold transistor source conductance, I/V_t , scales with input current, the gain, A , can be allowed to drop with higher input currents while still maintaining the effective low input impedance and stability. The AGC amplifier lowers its gain at higher output voltages, which correspond to larger input currents.

The log amp plays an integral role in the analog vector matrix multiplier (VMM), which performs the B matrix multiplication. As shown in Figure 5.5(c), every FGPFET in the array coupled with the respective row's log amp forms a wide-range, programmable gain current mirror. The current mirror utilizes the sources of the transistors for signal propagation instead of using the gates, as in [4], minimizing power law errors resulting from mismatches in gate-to-surface coupling. Each quadruplet of VMM FGPFETs corresponds to one coefficient in B . For a fully differential multiplication, w , the programmed gains for a quadruplet are set to $\begin{bmatrix} 1 + w/2 & 1 - w/2 \\ 1 - w/2 & 1 + w/2 \end{bmatrix}$. All VMM transistors along a row share the same input signal and perform their respective multiplications in parallel. The output currents are summed along the columns. The resulting differential current output vector is a vector-matrix multiplication vB . A similar single-ended structure is shown in [25], but does not emphasize low input impedance. Also, they use a current mirror on the front-end which introduces a possible kappa mismatch problem. A differential voltage-mode VMM is shown in [26], but does not have good dynamic range since it is built around voltage and not current multiplication. Current-mode techniques are usually required for processing wide dynamic range signals.

Lastly, a differential to single-ended I-V conversion structure, shown in Figure 5.5(c), was added to the back-end of the vector matrix multiplier. The output response is shown in Figure 5.5(d) with large dynamic range. The current subtraction which converts the differential signal to a single-ended signal is performed using a current mirror that also utilizes the source node for signal propagation. Though a gain error may occur because of threshold voltage mismatch, this is easily accounted for when programming the corresponding column of the VMM. Following the subtraction, a novel bidirectional current-to-voltage converter is used. This structure also utilizes a AGC amplifier, which loses gain as the output deviates from the zero current output voltage.

Figure 5.8 shows input vs. output current for a four-quadrant multiplier built with the

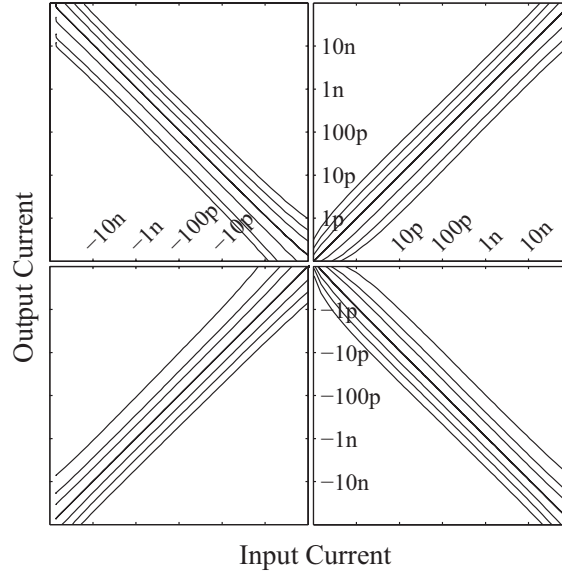


Figure 5.8. Multiplicative response of a programmable current mirror

programmable current mirrors described above. The four transistors were programmed at various levels to perform several multiplications. As shown, nominal operation over several orders of magnitude is possible. To precisely quantify the operational range, we programmed a single element of the current mirror and looked at the variance of $\frac{I_{out}}{I_{in}}$ over wide ranges. We were able to obtain a 2.5% error over three orders of magnitude at a multiplication of 1.5.

Figure 5.9 shows several preliminary results from the imager. Figure 5.9(a) shows a window view of a parking lot and parking structure, Figure 5.9(b) is the same image with a logarithm applied. The logarithm shows the dark window sill is captured in the same image with the bright outdoors. Figure 5.9(c) shows a 1-D DCT computed in the pixel plane and Figure 5.9(d) shows an ideal inverse DCT of that result. The successful reconstruction shows the correctness of the DCT computation. The log of the reconstruction, Figure 5.9(e), shows the range of input signal through the computation, which includes indoor and outdoor luminescence.

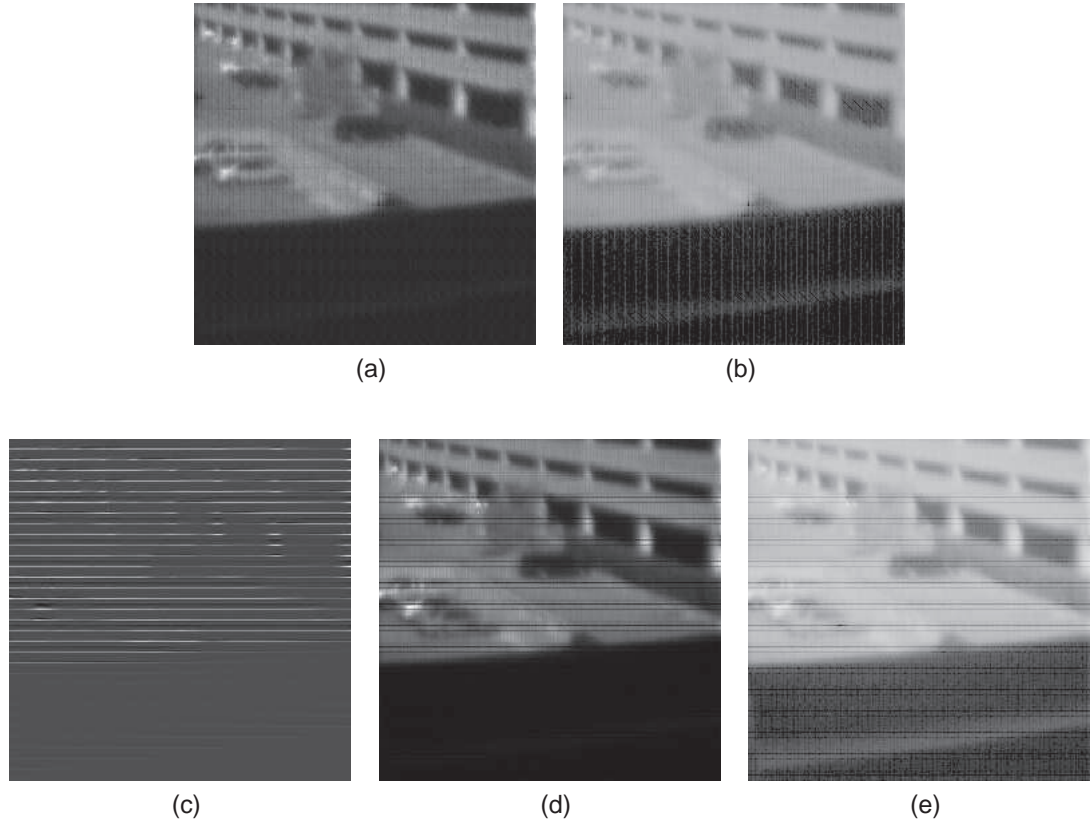


Figure 5.9. Preliminary image results of a parking lot and garage from a window view. (a) Identity Transform (b) Log of identity transform result (c) 1-D DCT computed in the pixel plan (d) Ideal inverse of DCT result (e) Log of inverse DCT result

5.4 Architecture Improvements

One issue with the previously mentioned architecture is that small currents are routed through multiple switches and along lengthy wires between the pixels and the logarithmic amplifiers located at the VMM. The switches introduce leakage currents and charge injection during digital control transistions. In the worst case, the transistions can cause momentary rapid decreases in current that the logarithmic amplifiers are not designed to handle. If the logarithmic amplifer can not respond by turning off its current in time, the line becomes overcharged to a high voltage. The recovery process involves discarging the line via the signal current, which can be small. Thus, these recovery times can be large. The last concern is that additional capacitive coupling from nearby wires can add and remove

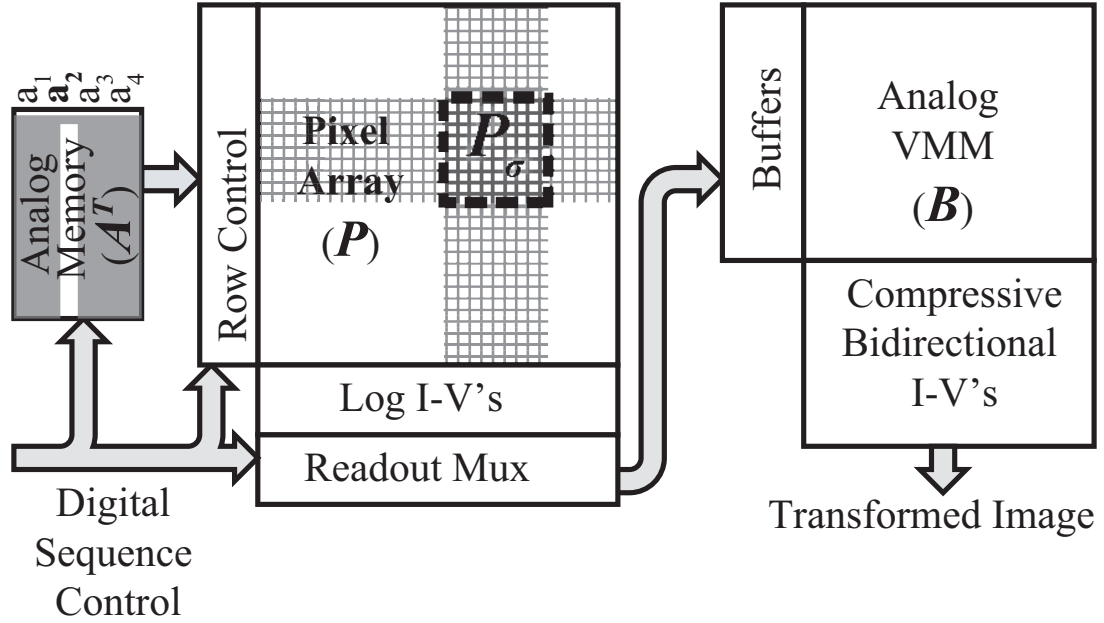


Figure 5.10. Computational imager sensor system level diagram showing the blocks of circuitry that implements the reprogrammable transform.

charges from these lines and effect the current measurements. These effects mentioned here are difficult or impossible to eliminate.

A better choice for signal propagation along such lengthy paths is voltage-mode signaling. To facilitate this, I moved the logarithmic amplifiers closer to the pixels in the most recent version of image sensor IC. Instead of being placed at the front-end of the VMM, as shown in Figure 5.1, they have been directly attached to pixel column outputs, as shown in Figure 5.10. This lowers the bandwidth requirements of the logamps, but they must now be placed at the output of every pixel column, requiring a compact design. In fact, because the system is differential, two are required per pixel column. Furthermore, the outputs of the amplifiers from different rows must be nearly perfectly matched to eliminate fixed, column-based offset patterns in the results.

A suitable design for for such an array of compact, matched logarithmic converters is shown in Figure 5.11. It includes a simplified, automatic-gain amplifier followed by a voltage buffer for driving the long output lines to the VMM. Of key importance in the voltage

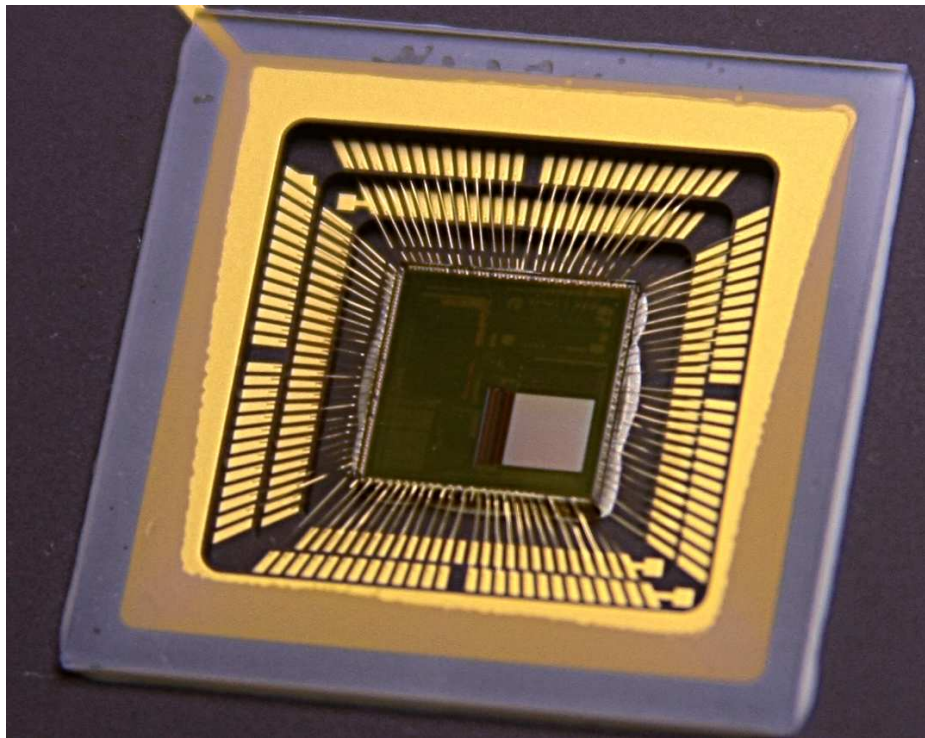


Figure 5.12. Newest image sensor IC die photo

CHAPTER 6

SENSING AND PROCESSING LOW CURRENT, WIDE DYNAMIC RANGE SIGNALS

One of the greatest challenges in sensing and processing is often dynamic range. Though many systems achieve acceptable SNR in certain signal ranges, they may not necessarily handle widely varying dynamic signals. Some systems incorporate tunable parameters to handle slowly varying DC and AC levels, but the real-world performance is often dictated by the ability to handle signals that vary on small timescales. Translinear techniques that use logarithmic compression to process signals are commonly used to process widely ranging signals. To our benefit, MOSFET transistors operating in the subthreshold regime inherently offer the ability to perform translinear operations, which can be used compress the signal before processing, since the voltage to current relationship involves an exponential. Using logarithmic representations of the signals, particularly in imaging, is a very natural way to handle signals. In fact, the human visual system is known to utilize logarithmic scales in the process of perception. A logarithmic conversion scales signal changes relative to the average signal value. The effect is that the absolute precision of the system is inversely proportional to the magnitude of the incoming signal so that relative precision is maintained. As a result, small signals are represented with enough precision and large signals are not represented with too much precision. As it turns out, relative signals, rather than absolute, are usually desired in most systems where wide ranges are involved. The difficulty in building such systems is in the trade-off between speed and power. Low currents usually coincide with slow speeds because of the presence of parasitic capacitances. Handling both low and high currents is particularly difficult because typical feedback systems require a power consumption proportional to dynamic range.

6.1 Programmable Subthreshold Current Mirroring

An examination of the basic current mirror in Figure 6.1(a) illuminates the points of concern when processing with subthreshold currents. The objective of a current mirror is to produce a current I_{out} which is proportional to the input current I_{in} . This is done by cascading a current to voltage conversion and a voltage to current conversion. First, a current is pulled through the input transistor and the voltage V_g settles at a point that satisfies the I-V relation of the transistor.

$$V_g = \frac{U_t \ln\left(\frac{I_{in}}{I_0}\right) + V_s}{\kappa_1} \quad (6.1)$$

Applying this gate voltage to the second transistor yields

$$I_{out} = I_{0,2} e^{\frac{\kappa_2 \left[\frac{U_t \ln\left(\frac{I_{in}}{I_{0,1}}\right)}{\kappa_1} \right]}{U_t}} \quad (6.2)$$

which simplifies to

$$I_{out} = k I_{in}^{\left(\frac{\kappa_2}{\kappa_1}\right)} \quad (6.3)$$

where $k = \frac{I_{0,2}}{I_{0,1}^{\left(\frac{\kappa_2}{\kappa_1}\right)}}$. Attention to the exponent $\frac{\kappa_2}{\kappa_1}$ in Equation 6.3 is important when working in a system with large dynamic range. Values of the ratio other than unity cause a power law relationship with errors that grow disproportionately to the input signal. For example, assume that α could be controlled such that at some input current, $I_{in} = I_{in0}$, the output current is perfectly matched, $I_{in} = I_{out}$. This requires $\alpha = I_{in0}^{\left(1 - \frac{\kappa_2}{\kappa_1}\right)}$. However, when $I_{in} = m \times I_{in0}$, the result is $I_{out} = m^{\frac{\kappa_2}{\kappa_1}} \times I_{in0} = I_{in} \times (m^{\frac{\kappa_2}{\kappa_1} - 1})$. So, even if κ_1 and κ_2 match with 1% error, the output current has 4.71% error with $m = 100$, two orders of magnitude.

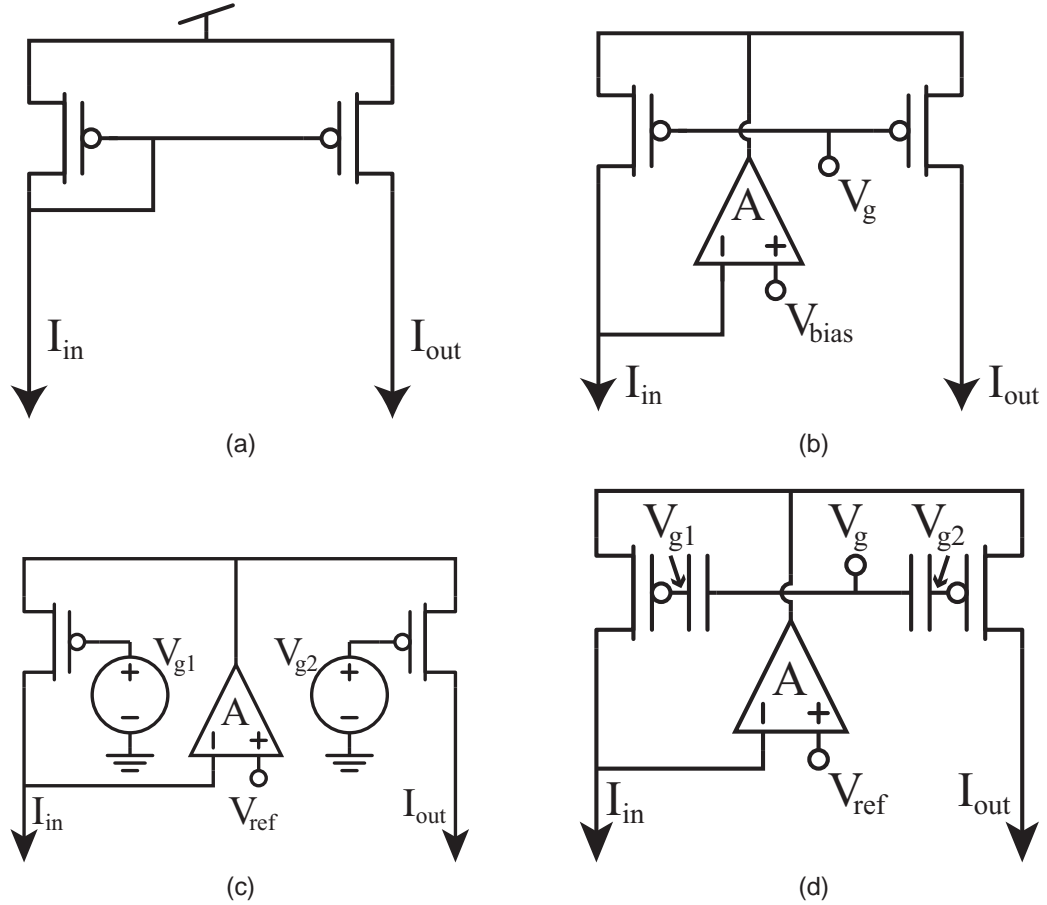


Figure 6.1. Current mirrors(a) Simple current mirror utilizing the gate voltages to mirror the current. (b) Active current mirrors utilizing the source voltage to mirror currents. The amplifier creates a high gain feedback loop which speeds the response at the input node while providing drive strength for the source nodes. (c) A tunable gain, subthreshold current mirror. The gain is set by the difference $V_{g1} - V_{g2}$. This structure allows reprogrammable gain and mismatch compensation. Utilization of source voltage variation to mirror the current avoids the power law mismatch due to kappa variance between the two transistors. (d) A floating-gate programmable gain subthreshold current mirror, utilizing built-in storage.

One solution to kappa mismatch is to utilize a structure which does not rely on kappa matching. Figure 6.1(b) shows such a structure which utilizes the source instead of the gate for signal conveyance. In this structure the input-output current relationship is as follows:

$$I_{out} = kI_{in}; k = \frac{I_{0,2}}{I_{0,1}} e^{\frac{(\kappa_1 - \kappa_2)V_g}{U_t}} \quad (6.4)$$

Here, we do not incur the power law, only a constant multiplicative error set by the subtraction $\kappa_1 - \kappa_2$ and the ratio $\frac{I_{0,2}}{I_{0,1}}$, which is caused by mismatches in transistor sizes and threshold voltages. By creating a voltage difference on the gates of the transistors, as shown in Figure 6.1(c), a multiplication (or division) can be set as follows:

$$I_{out} = kI_{in}; k = \frac{I_{0,2}}{I_{0,1}} e^{\frac{\kappa_1 V_{g1} - \kappa_2 V_{g2}}{U_t}} \quad (6.5)$$

Equation 6.5 shows that the two gate voltages can be adjusted to compensate for mismatches in sizes and threshold voltages, while also providing a desired multiplication. An exploration of this topic is given in [28], where a variety of structures with input clamping and tunable gain based on applied control voltages are shown.

Floating-gate transistors offer another particularly flexible option for setting the gain of the transistors. In the implementation shown in Figure 6.1(d), the offset voltages can be programmed as charges on the floating node associated with each mirror transistor. This implementation avoids the requirement of a unique voltage source and buffer for every gate, which is very cumbersome for large arrays.

The implementation of large arrays of tunable or programmable mirrors imposes certain restrictions on the performance of the mirrors. In general, capacitor and/or a buffer is used to steady the gate voltage, but the overlap capacitance between the source and drain will couple the varying source voltage onto the gate. This causes undesired fluctuations on the gate. The coupling can be described by the following equation:

$$\frac{v_g}{v_s} = \frac{C_{ov}}{C_T} \frac{sRC_T}{1 + sRC_T} \quad (6.6)$$

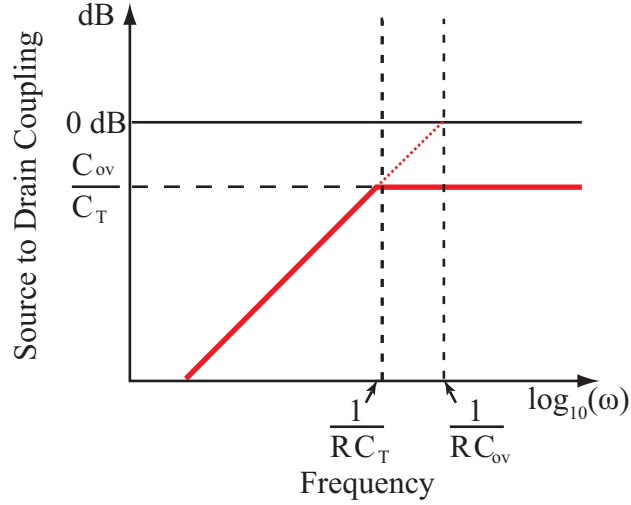


Figure 6.2. Source to gate coupling. The effect of the source on the otherwise fixed gate is dependent on the frequency, gate-to-source overlap capacitance, total gate node capacitance, and the DC resistance to ground set by any amplifiers driving the node.

R is ∞ for a floating gate since no connection is made to the gate. In the case where the gate node is driven with a source-follower, R is $1/g_s$. In the case it is driven by an amplifier with unity gain feedback, R is $1/g_m$. Figure 6.2 is a useful visualization for interpreting this expression. Below the corner set by the node resistance and the total node capacitance, $\omega_1 = \frac{1}{RC_T}$, the coupling is frequency dependent, potentially limiting the operational speed of the structure. The magnitude of the voltage movement is determined by transfer curve set by the node resistance and the overlap capacitance, $\omega_2 = \frac{1}{RC_{ov}}$. Larger values for ω_2 shift the curve to the right, lowering the response at the lower frequencies. Above the frequency ω_1 , the coupling is limited by ratio of the overlap capacitance to the total capacitance. For a floating-gate transistor, the corner occurs at 0 Hz, so the transfer function is a constant C_{ov}/C_T . In summary, the overlap capacitance should be small compared to the conductance, $1/R$, setting the node, or it should be small compared to the total node capacitance. The same issue exists for the drain, but the assumption here is that the drain is held fixed by whatever circuit is sinking or sourcing the current output.

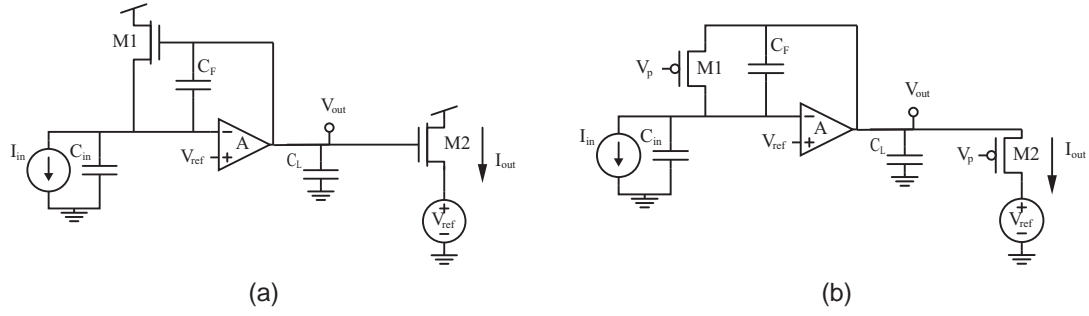


Figure 6.3. Logarithmic transimpedance amplifier topologies. (a) common drain (b) common gate

6.2 Logarithmic Transimpedance Amplifiers

Figure 6.3 shows two topologies for logarithmic transimpedance amplifiers (logamps). In both structures, transistor M1 is kept in subthreshold operation. This enables the logarithmic conversion from input current I_{in} to output voltage V_{out} . The transfer function of the common-drain topology in Figure 6.3(a) is:

$$\Delta V_{out} = \frac{1}{\kappa/U_t + 1/AU_t} \ln\left(\frac{I_{in}}{I_p'}\right) \approx \frac{U_t}{\kappa} \ln\left(\frac{I_{in}}{I_p'}\right) \quad (6.7)$$

The transfer function of the common-gate topology in Figure 6.3(b) is:

$$\Delta V_{out} = \frac{1}{1/U_t + 1/AV_A} \ln\left(\frac{I_{in}}{I_p'}\right) \approx U_t \ln\left(\frac{I_{in}}{I_p'}\right) \quad (6.8)$$

For comparison, Figure 6.4 shows a simple logarithmic current-to-voltage converter. It has good output voltage driving capability, but poor (large) input resistance which could be inadequate for large input capacitances and small input currents. Its transfer function is as follows:

$$\Delta V_{out} = U_T \ln\left(\frac{I_{in}}{I_p'}\right) \left(\frac{A}{1+A}\right) = U_T \ln\left(\frac{I_{in}}{I_p'}\right) \left(\frac{1}{1+1/A}\right) \approx U_t \ln\left(\frac{I_{in}}{I_p'}\right) \quad (6.9)$$

The approximations assume that the gain A is very large. The input-referred error of this approximation is manifested when using the output voltage to mirror the inputted current

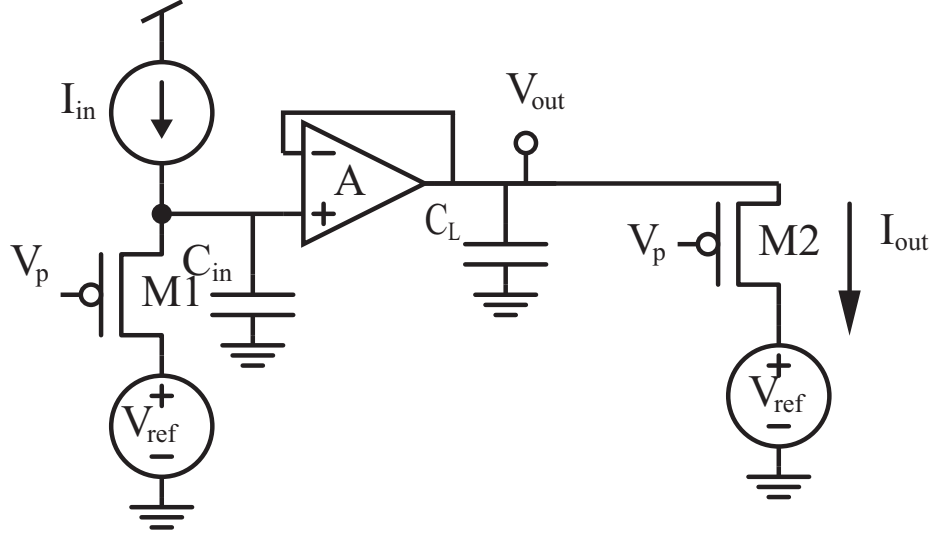


Figure 6.4. Simple I-V. Output drive is provided, but not a low input resistance.

in M2 over several orders of magnitude of current. This can be calculated using the full expressions for changes in voltage given in equations 6.7 and 6.9, and applying those to the gate of an NFET and source of a PFET respectively.

If the equations for the outputs of the circuits in Figures 6.3 and 6.4 are placed into the form $I_{in} = c(I_{in})^{\frac{1}{1+\alpha}}$, their performances with different values for A , the internal amplifier gain, can be compared. Assuming $\alpha \ll 1$, the following expression describes the error introduced when the input current changes over a range bounded by I_{in0} and $m \times I_{in0}$.

$$\%Error = -\alpha \ln(m) \times 100 \approx -\alpha \times 230 \times \log_{10}(m)$$

The values of alpha are $\frac{U_T}{AV_A}$ for the common-gate logamp and $\left(\frac{\kappa_1}{\kappa_2} - 1\right) + \frac{U_T}{\kappa_2 AV_A}$ for the common-drain logamp. For comparison, the buffer in the simple I-V converter in Figure 6.4 creates an alpha of $1/A$. The common-drain logamp parameter includes the effect of kappa mismatch in transistors. Even assuming the kappa values matched perfectly, the common-gate amplifier would still be better than the common-drain configuration by a factor of $1/\kappa$ for the same amplifier gain. Both logamps provide low input resistance and can

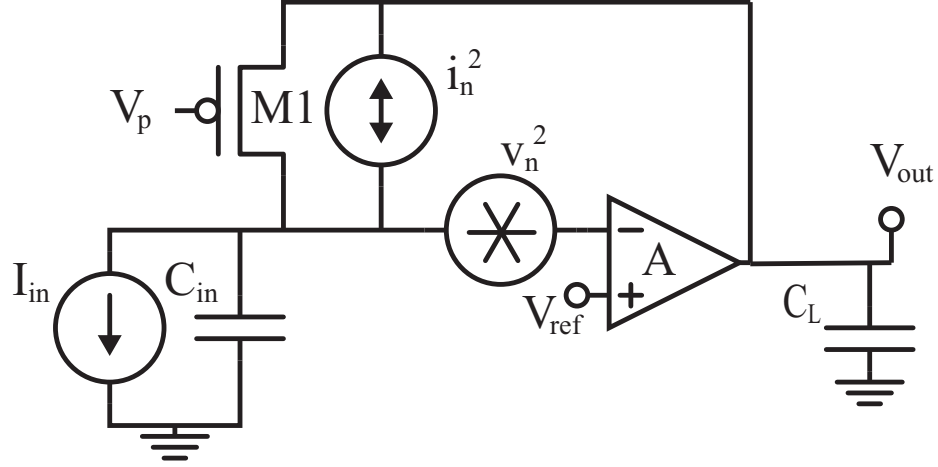


Figure 6.5. Logarithmic transimpedance amplifier noise sources.

sense low pixel currents on the readout lines. There are many trade-offs between the two designs, but a particular advantage of the common-gate topology is the upper bound on speed. The Miller-effect of the gate-to-source capacitance in the common drain configuration limits the achievable bandwidth to $1/(U_i C_{gs})$. The Miller-effect is the effective multiplication of a capacitor by the gain of an amplifier when that capacitor is placed in the negative feedback path across that amplifier. We chose to use the common-gate logamp because of the superior speed and accuracy when used in a current mirror.

6.2.1 Noise

When designing a logarithmic amplifier, one needs to consider the noise contribution of the internal amplifier and the noise of the feedback element. There is a near continuum of design variations depending on the requirements of the systems, but for this discussion we will consider that the internal amplifier is acting as a voltage amplifier with a single pole, $A \frac{1}{1+s\tau_A}$, independent of the feedback current. As shown in Figure 6.5, which depicts a common-gate feedback topology, there are two noise components to consider: the lumped amplifier voltage noise, v_n^2 , and the transistor current noise, i_n^2 . A full expression for output

noise can be derived as

$$V_{out} = G \frac{i_{in} \frac{1}{g_s} + v_n \frac{1}{\frac{g_s}{g_d} \left(\frac{1}{1+s\frac{C}{g_d}} \right)}}{-s^2 \frac{C}{A g_s} \tau_A G - s \left(\frac{C}{A g_s} + \frac{\tau_A}{A \frac{g_s}{g_d}} \right) G + 1}; G = \frac{A \frac{g_s}{g_d}}{1 - A \frac{g_s}{g_d}} \approx -1 \quad (6.10)$$

Referring the noise to the input gives another view, related to the level of input signal that can be sensed:

$$i_{in} = i_n + v_n \left(g_s \frac{1}{\frac{g_s}{g_d} \left(\frac{1}{1+s\frac{C}{g_d}} \right)} \right) \quad (6.11)$$

For comparison, a common-drain topology is given here:

$$i_{in} = i_n + v_n \left(g_s \frac{1}{\frac{g_m}{g_s} \left(\frac{1}{1+s\frac{C}{g_s}} \right)} \right) \quad (6.12)$$

In equations 6.11 and 6.12 the contributions of the noise components, i_n and v_n are seen. The noise from the feedback transistor, i_n , appears like a noise source at the input node in parallel with the input signal. The noise from the amplifier, v_n , is attached to a more interesting expression.

The noise from the amplifier is divided down by the frequency-dependent gain of the feedback element. While this might sound advantageous, one must realize that the feedback gain is likely less than 1 at the frequency of interest. It is illustrated in Figure 6.6(a) that the gain of the feedback element drops below 1 at a certain point. In fact, a major purpose of the negative feedback loop is to boost the speed of operation beyond the unity gain frequency of a single transistor. Since the amplifier noise is divided by the feedback gain, the inverse gives the amplifications. Figure 6.6(b) shows that past a certain point, the amplifier's noise is multiplied up by values greater than 1. So, the amplifier should be designed with this in mind.

Even if the amplifier could be designed so that its noise contribution was negligible, the noise contribution from the feedback transistor remains as an inherent characteristic of the system. The noise in this device arises from the same statistical randomness that causes

noise in the current being measured. So, it and the input source should be considered. In the end, for low currents and fast speeds, one faces the physical phenomenon of quantized charge movement. More insight can be drawn by thinking of the current measurement task as electron counting. The fundamental problem can be realized by treating the current as the movement of discrete quantities of charge, q , according to a Poisson process with parameter λ .

In a given time, T , with an average current level, I , the number of carriers, n , passing a point is as follows:

$$n = \frac{I}{q} \times T \quad (6.13)$$

A Poisson process has a variance and mean that are equal. They are given by a characterizing parameter λ . Therefore, in this case, the characteristic parameter is

$$\lambda = \frac{I}{q} \times T \quad (6.14)$$

If we take the SNR to be the mean divided by the standard deviation, we have a simple result:

$$SNR = \frac{\mu}{\sigma} = \frac{\lambda}{\sqrt{\lambda}} = \sqrt{\lambda}. \quad (6.15)$$

Simply stated, if the desired SNR is 100, we can not measure a current in less time than it takes 100^2 electrons to pass. For any current- measurement circuit the following equation gives a lower bound on the sampling period as a function of current and SNR:

$$T_s = SNR^2 \times \frac{q}{I} \quad (6.16)$$

For an SNR of 100, or 40dB, and 1nA current, the maximum sampling frequency is 624KHz. In the case where the input signal is in the presence of a large offset current, the SNR can still defined in terms of the input signal if the sampling speed equation is modified

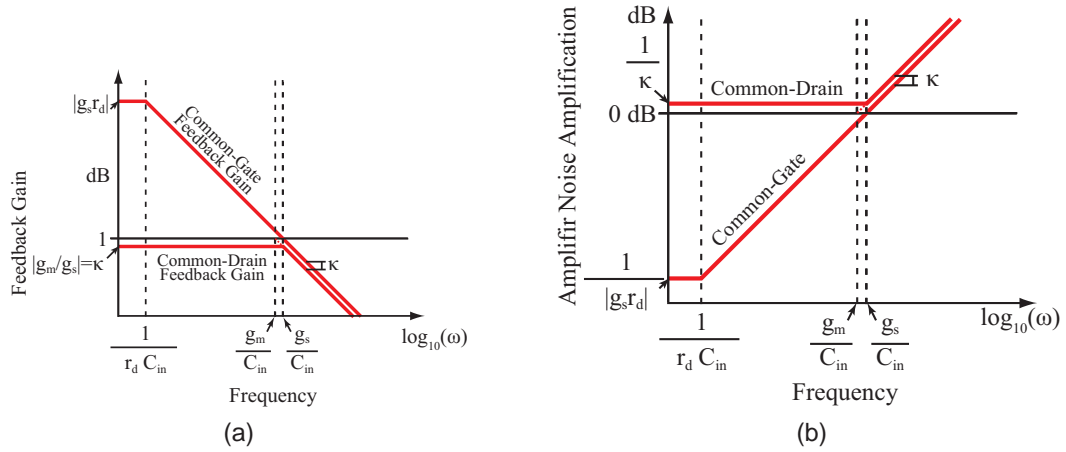


Figure 6.6. Logarithmic amplifier feedback element gain. (a) Feedback element gain for common-gate and common-drain topologies. (b) The inverse shows the gain affecting the internal amplifier's noise contribution, since the noise of the amplifier is divided by the feedback gain. In the area of interest, beyond the unity gain frequency of the feedback devices, the common-gate has a small advantage.

to include the offset current:

$$f_s < \frac{I_{signal} + I_{offset}}{q \cdot SNR^2 \left(\frac{I_{signal} + I_{offset}}{I_{signal}} \right)^2} = \frac{I_{signal}}{q \cdot SNR^2} \times \frac{I_{signal}}{(I_{signal} + I_{offset})},$$

So, to detect 1 nA signal with 1% accuracy in the presence of a 100 nA offset, the achievable frequency is limited to 6.18 kHz. Attempts to remove the offset current by sampling it and producing a canceling current at the input would not help, since the canceling currents would contribute the same noise. Keeping the offset currents small from the beginning is, therefore, essential in such low-current sensing architectures.

6.2.1.1 Power Dissipation

Considering the logamp as a two pole system and analyzing power requirements, the required gain of the amplifier is determined by the desired bandwidth, the minimum input current, and the input capacitance:

$$A \frac{I_{in,min}}{U_T} \frac{1}{C_{IN}} > BW_{min}, \quad (6.17)$$

With the input node setting the dominant open-loop pole, we require the second open-loop pole to be large enough for stability in the closed-loop system.

$$p_{2, \text{OPENLOOP}} > BW_{\max} = \frac{AI_{\text{in}, \max}}{U_T C_{\text{IN}}}. \quad (6.18)$$

So, transconductance in the amplifier must obey the requirement

$$G_m > \frac{U_T C_L C_{\text{IN}} (BW)^2 DR}{I_{\text{in}, \min}}. \quad (6.19)$$

Equation 6.19 is very important because it shows the dependence of the power dissipation on the design parameters. One can correlate power consumption with the required transconductance G_m . In subthreshold operation, transistor transconductances are linearly proportional to current and power. In above-threshold operation, the transconductance is proportional to the square root of the current and power. Therefore, the power is proportional to dynamic range, input and output capacitances, and the square of bandwidth, but is inversely proportional to the minimum input current. Dynamic range is defined here by

$$DR = \frac{I_{\text{in}, \max}}{I_{\text{in}, \min}} \quad (6.20)$$

The dependence on dynamic range can be reduced if the amplifier is made to be adaptable. One method examined was the lowering of output resistance, and consequently gain, at higher input currents. Also examined was increasing the G_m dynamically by changing the tail current in the amplifier, though in worst case, when the input currents are high, this does not yield any power savings. Theoretically, a perfect compensation scheme that adjusts output resistance inversely to input current would completely remove the dynamic range power dependence.

In the chosen design, we a multiple stage amplifier was used. Multiple stages often save power but introduce additional noise. In the amplifier, the gain is adjusted using

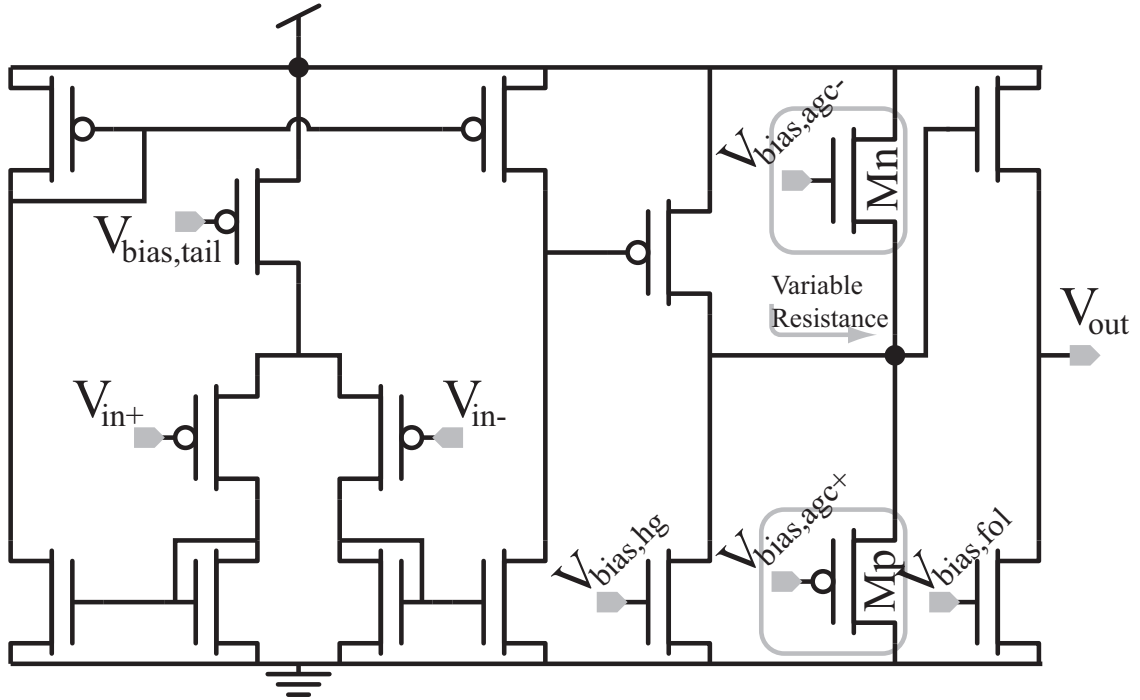


Figure 6.7. Dynamic amplifier. This amplifier will vary its gain as the output moves away from the tunable point of maximum gain. This is used in both the current mirrors and the bidirectional I-V converter to produce an automatic gain control dependent on the input current levels. The gain is highest when the inputs currents are small, providing the most loop gain and speedup, and lowest when the currents are high, minimizing the power requirements for stability.

an automatically varying resistance that is internal to the amplifier. Figure 6.7 shows the design. The two variable resistances are seen looking into the sources of Mn and Mp. The maximum gain point can be adjusted by moving the bias voltages at the gates of the transistors.

6.3 Bi-Directional Compressive Transimpedance Amplifier

Bidirectional current-to-voltage conversion is a particularly difficult problem. The single-ended approaches discussed relied on some amount of input current flow to operate. If the input current were to go to zero, the feedback transistor would turn off, leading to very slow operation. In the bidirectional case, the converter must be able to operate with no input current since the current must pass through zero as it changes direction.

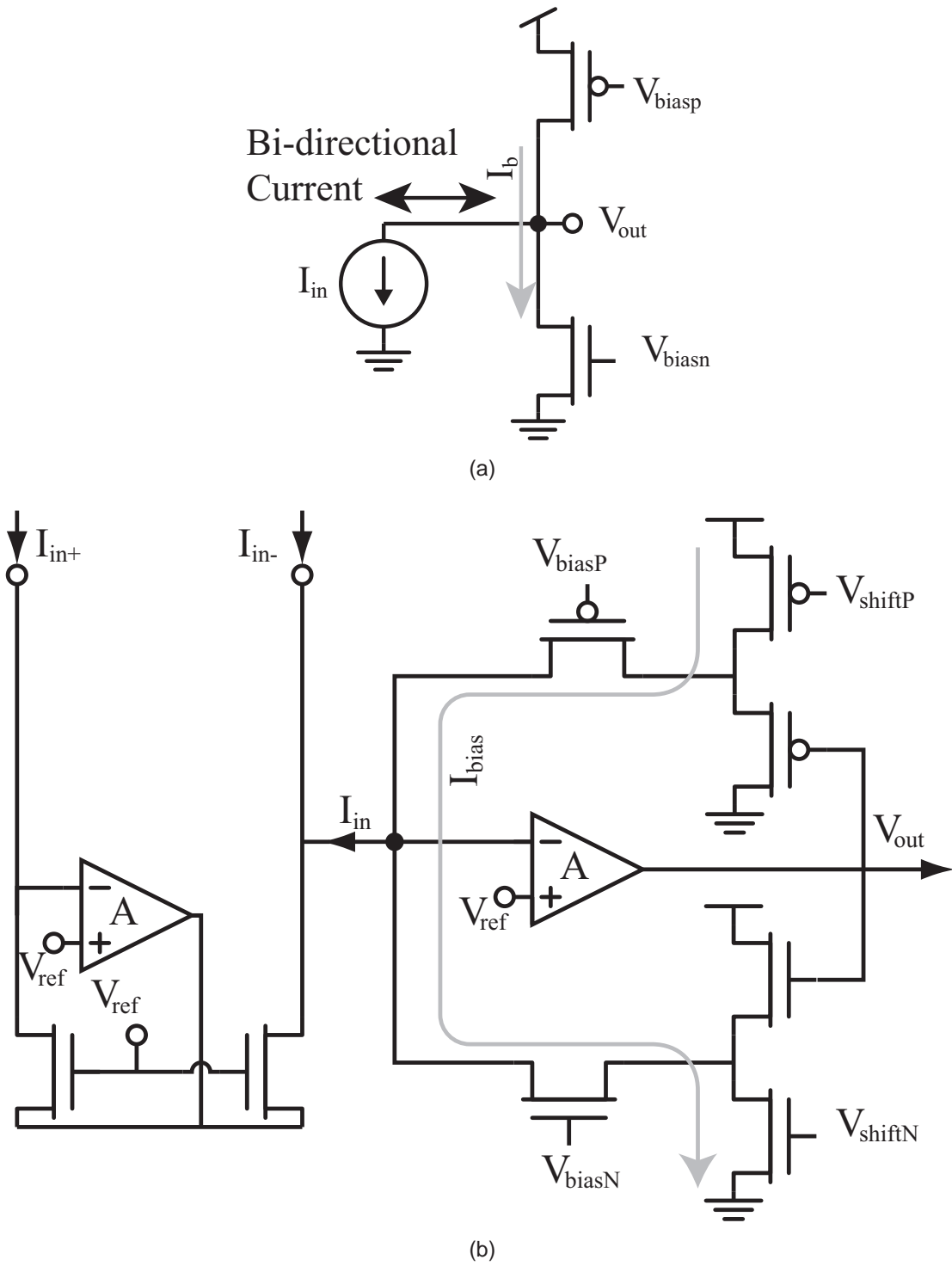


Figure 6.8. Bidirectional I-Vs. (a) Simple Compressive I-V (b) High-speed, low-current differential-to-single-ended I-V converter

In Figure 6.8(a), a simple circuit is shown which converts a bidirectional input current to a voltage. The structure is biased such that there is always a bias current I_b flowing through the devices, eliminating the speed concern at zero input current. Utilizing subthreshold transistor exponential characteristics, one can write the input current as

$$I_{in} = -I_b(e^{-\Delta V_{out}/U_t} - e^{\Delta V_{out}/U_t}) = 2I_b \sinh(\Delta V_{out}/U_t) \quad (6.21)$$

Solving for ΔV_{out} we get

$$\Delta V_{out} = U_t \times \sinh^{-1}\left(\frac{I_{in}}{2I_b}\right) \quad (6.22)$$

I_b must be chosen to satisfy both the converter's sensitivity requirements and the converter's minimum speed requirements. This creates a trade-off since the sensitivity is inversely proportional to I_b while the minimum speed is directly proportional to I_b . This is because the bandwidth of a subthreshold transistor, $\frac{g_m}{C}$, is directly proportional to I .

To achieve higher speeds at low currents, feedback was used to lower the input resistance. The circuit is shown in Figure 6.8(b). The structure operates in a similar fashion to the common-gate transimpedance amplifier, utilizing the exponential current-to-voltage relationships at the sources of the feedback transistors. The source followers introduce the appropriate offsets so that when the input current is zero, there is still a bias current running through the feedback transistors. Without this offset, the transistors would have to operate without source-to-drain current when the input current is near zero, rendering them extremely slow. Higher bias currents increase the speed of operation at lower input currents, but reduce the low-current resolution, just as in the non-feedback counterpart. The followers act to provide appropriate source voltages to each feedback transistor. The NMOS and PMOS transistors share their drain terminal. The NMOS requires a source voltage that is lower than the drain voltage and the PMOS requires a source voltage that is above the drain voltage. Without the individual level-shifting source followers, these conditions could not be both satisfied for all output currents.

This structure has a similar transfer characteristic to the non-feedback version, but the sign is negated. In addition, because the followers have different gains, $\kappa_{p, \text{follower}}$ and $\kappa_{n, \text{follower}}$, an asymmetry is introduced in the transfer characteristic. The complete transfer characteristic is

$$I_{in} = I_b \left(e^{\frac{\kappa_{p, \text{follower}} \Delta V_{out}}{U_t}} - e^{\frac{-\kappa_{n, \text{follower}} \Delta V_{out}}{U_t}} \right) \quad (6.23)$$

It can be approximated as two separate functions when $\Delta V_{out} \gg 0$,

$$I_{in} = I_b \left(e^{\frac{\kappa_{p, \text{follower}} \Delta V_{out}}{U_t}} \right) \quad (6.24)$$

and when $\Delta V_{out} \ll 0$,

$$I_{in} = I_b \left(-e^{\frac{-\kappa_{n, \text{follower}} \Delta V_{out}}{U_t}} \right) \quad (6.25)$$

Thus, as the input current becomes large, the converter approximates a logarithmic compression. This bi-directional converter is very useful in applications where support for large dynamic range is essential and small currents must be sensed at bandwidths well beyond g_m/C .

CHAPTER 7

MISMATCH AND OFFSET REMOVAL

An understanding of the non-idealities of this or any system is crucial to effective system utilization and possible compensation of errors. In this pixel plane, there is a massive collection of parallel multiply-accumulate cells. Unfortunately, each device varies, introducing undesirable errors. The sources of error include transistor threshold mismatches, photodiode mismatches, and parasitic light-sensitive junctions.

7.1 Pixel Array Characteristics and Mismatch

Column offsets are common in imager architectures, including active pixel sensor (APS) imagers [29]. In APS imagers, the column offsets have been attributed mostly to offsets in column amplifiers. Fixed pattern noise is treated as a combination of a column offset and pixel offsets. In our imager, the distinction is that the individual pixel offsets create a large cumulative contribution to the column offsets. The column readout circuitry still creates additional offsets. This and other parasitic effects common to column readout lines can cause offsets, which are most observable under uniform illumination exposure.

Figure 7.1 shows the extraction of I_{mid} , as described in section 4.2, for a two-dimensional pixel array. The column effects are clearly visible here. To understand and then remove these errors, a series of experiments were performed. This resulted in attributing the source of column errors to the aggregate error of the pixels on the column.

If mismatches in the threshold voltages of the two transistors occur, a horizontal voltage offset of the curve results. W/L mismatches have much less of an effect than threshold voltage V_t , since V_t is in an exponential along with V^+ and V^- in the subthreshold model, while W/L is not. The voltage offset is multiplied by the negative transconductance of the differential pair, $-G_m$, to produce an offset current. The offset currents are aggregated along the column lines to produce the most significant portion of I_{mid} . The inverse correlation of

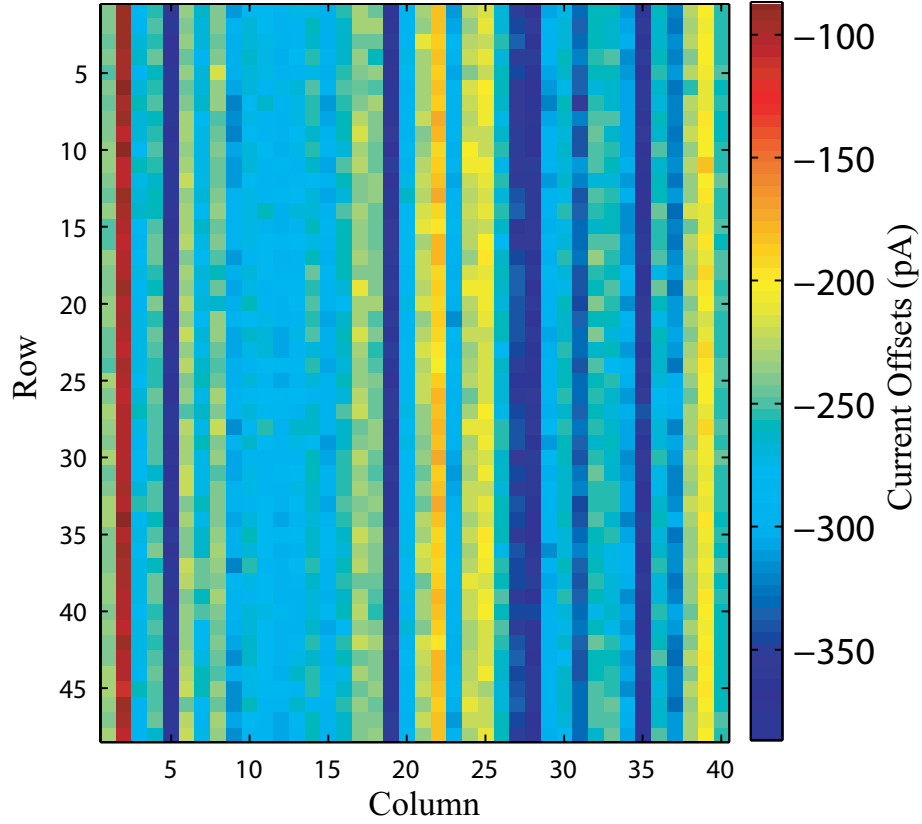


Figure 7.1. Current offsets showing large column striations (column offsets)

I_{mid} and V_{offset} can be seen in Figure 7.2, which shows the mean voltage offsets and mean current offsets for each column of a pixel array. The correlation is not perfect because there are other factors in the column offsets that are described later.

There are several parasitic reverse-biased diode junctions along the column line that exhibit leakage current. These junctions are subjected to light, which means that they act as parasitic photodiodes. The combination of parasitic photodiodes and the voltage offsets of each pixel creates an image-dependent offset in each column. It is dependent on the image because the amount of light falling on each pixel and each parasitic junction determines the contributions to the column offset. Image dependence simply means the offset will not be constant. This makes removing it more difficult than simply subtracting a constant from each column or applying a scale factor.

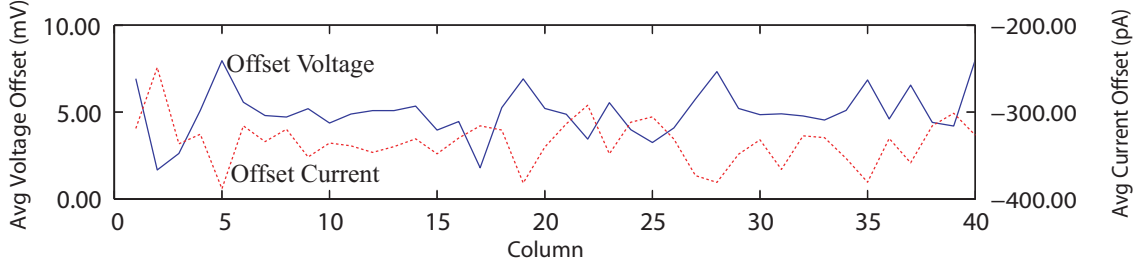


Figure 7.2. Average column voltage offsets and column current offsets. As expected positive voltage offsets correlate with negative current offsets.

Also affecting results in the characterization chips used were electrostatic discharge (ESD) protection on the output lines. ESD protection was implemented using reverse-biased diodes to power and ground. These reverse-biased diodes unfortunately act as large photodiodes and cannot be covered by metal. To reduce the effects of the diode protection, later characterization chips moved the diode protection away from the edge of the chip so that they could be better shielded from light using top-level metal layers.

The next parameter for discussion is the gain in the linear region, denoted by G_m in Figure 4.3. From Equation 4.7 we see that the gain term is simply

$$Gain = I_{photo} \frac{\kappa}{2U_t} \quad (7.1)$$

Also, note that kappa can be obtained using

$$\kappa = Gain \times \frac{2U_t}{I_{photo}} \quad (7.2)$$

I_{photo} can be found experimentally by using the fact that the height of the tanh curve is $2 * I_{photo}$. Taking the difference of the two extremities of the tanh curve gives us the value needed to solve for kappa with the assumption that U_t is based on room temperature.

To measure the variation of parameters over an array of pixels, individual I-V sweeps were taken. The extracted parameters are shown in Figures 7.3, 7.4, 7.5, and 7.6.

Figure 7.3 shows the gain across an array under nearly uniform illumination. Edge

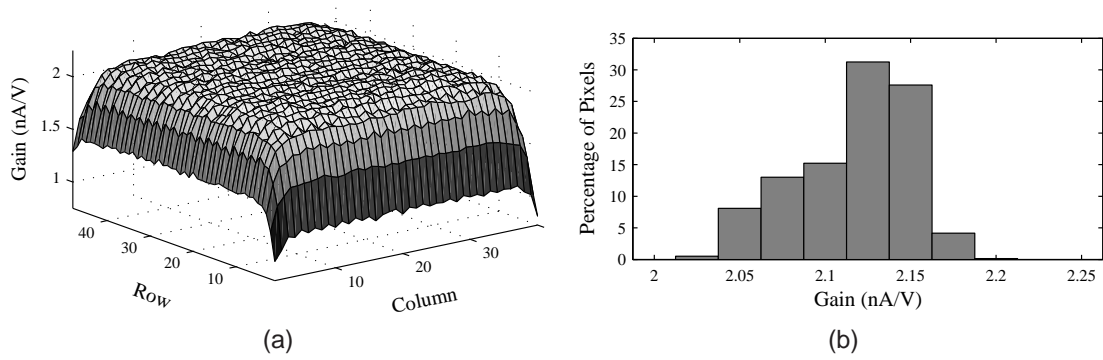


Figure 7.3. Gain mismatch. (a)Gain as a function of pixel position. (b)Histogram of gains (outer 8 pixels are excluded from statistics)

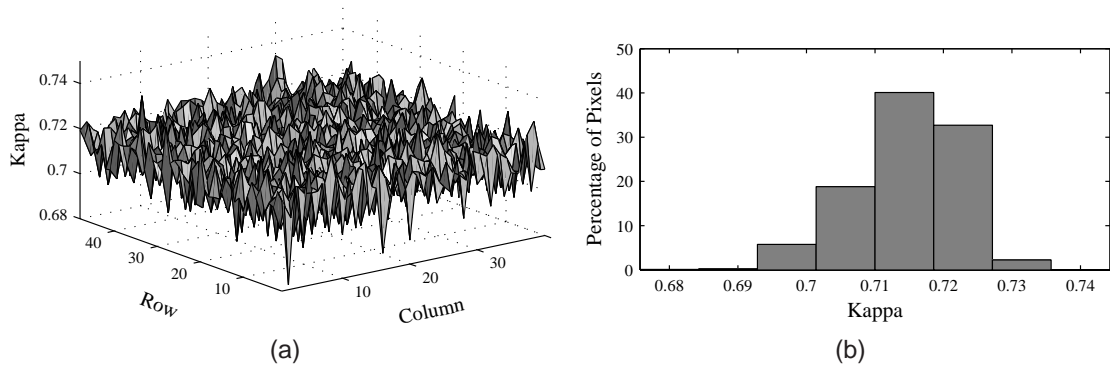


Figure 7.4. Kappa mismatch. (a)Kappa as a function of pixel position. (b)Histogram of kappa

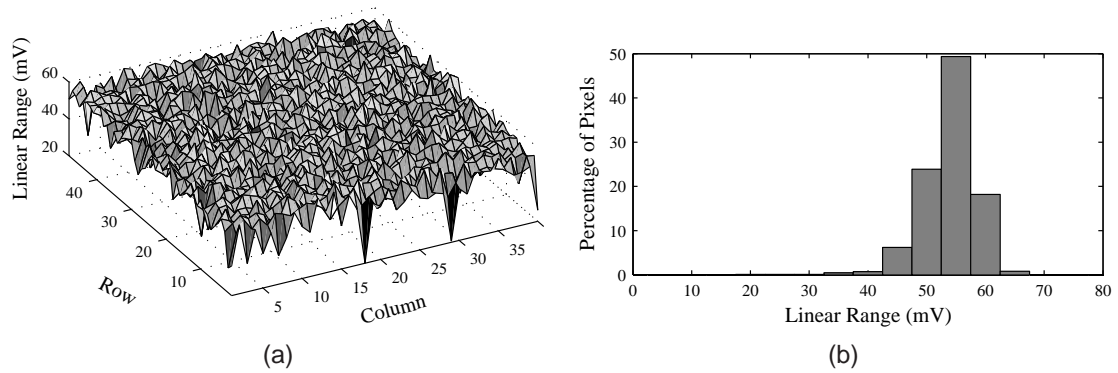


Figure 7.5. Linear range. (a)Linear Range as a function of pixel position. (b)Histogram of linear ranges

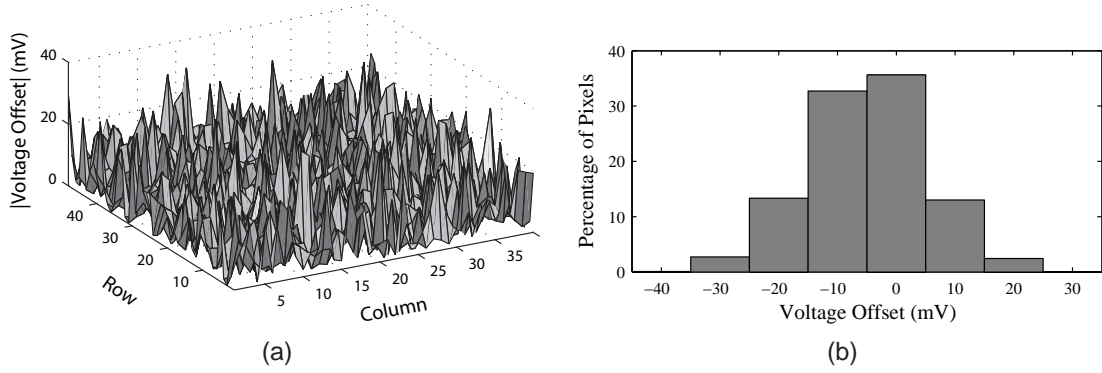


Figure 7.6. Voltage offsets. (a)Absolute voltage offsets of differential pairs as a function of pixel position. (b)Histogram of voltage offsets

Table 7.1. Pixel statistics extracted from a pixel array

	Mean	Std. Dev.
Gain	2117.4 pA/V	33.6 pA/V
Linear Range	54.4 mV	4.3 mV
V_{offset}	4.9 mV	10.0 mV
V_{offset}	8.9 mV	6.7 mV
Kappa	0.715	0.007

effects characteristic of CMOS imagers are clearly seen as in other array characterizations[30]. Since pixels near the edge of the array have different physical surroundings than the pixels toward the middle, they tend to have variations. The gain mismatch seems to originate from variations in the photodetector current, as seen in Figure 7.10. The edge effect does not always show a falloff, and different edges on the same imager may show different characteristics. Edge effects did seem consistent among chips on the same process run of the same design. There seemed to be no edge effects in the kappa measurements, suggesting that the effect occurs in the photodiode itself and not in the transistors. So, the gain error is caused by mismatch of photosensor size and efficiency and also kappa. Overall, though, the gained seemed to be within usable margins of error. Table 7.1 shows the extracted statistics, which were taken from the center of an array to exclude edge effects.

Moving on to discuss voltage offset measurements, the results in Figure 7.6 and Figure 7.7 show voltage variations mostly in a $\pm 30mV$ range, as expected. A normal distribution

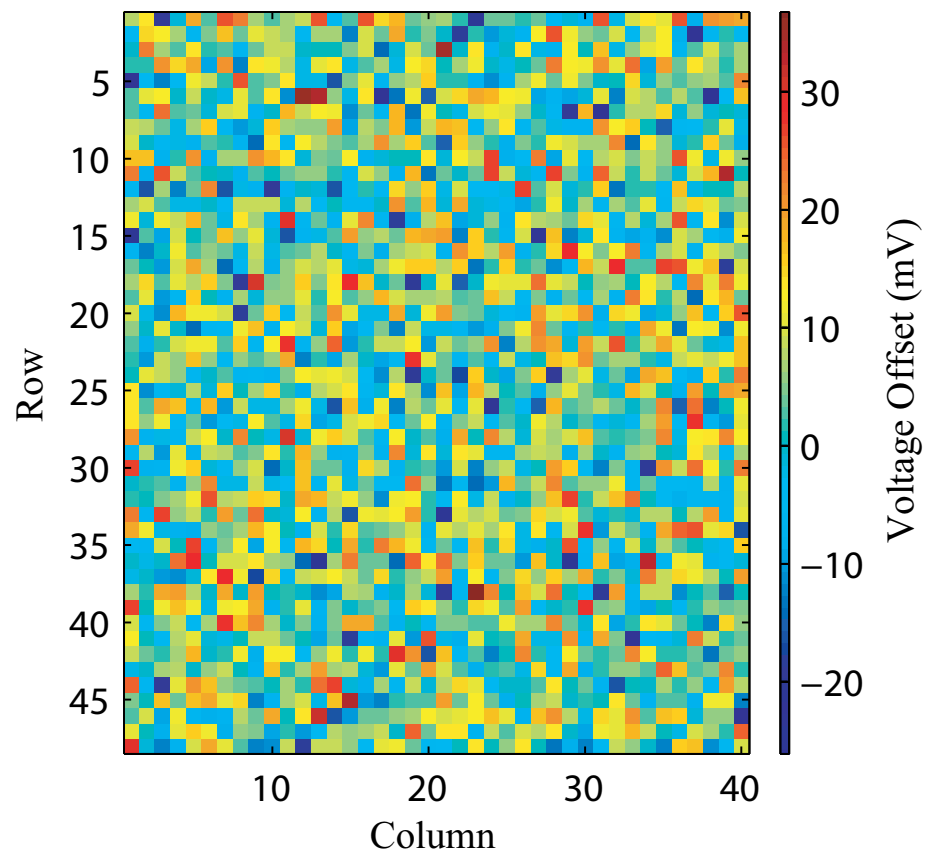


Figure 7.7. Voltage as a function of position, showing a mostly random distribution of voltage offset. Spatially random effects dominate any gradients that may be present

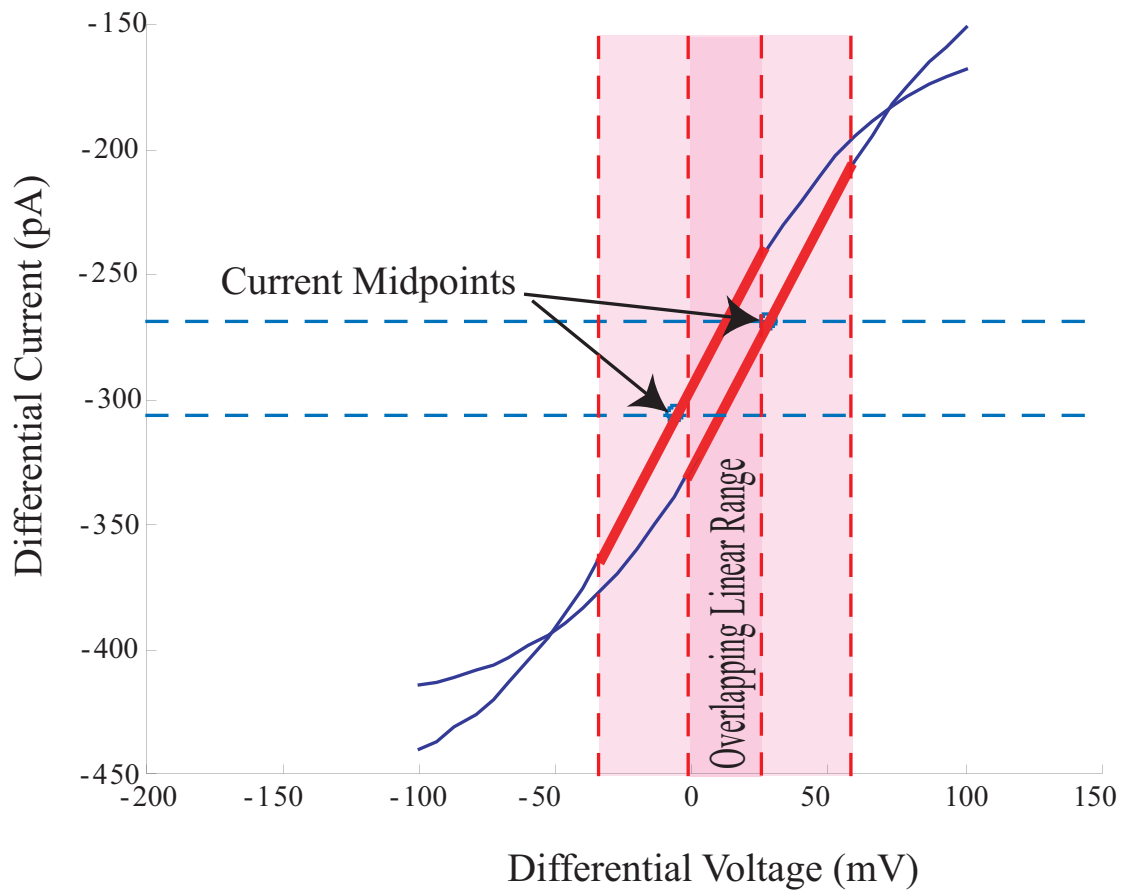
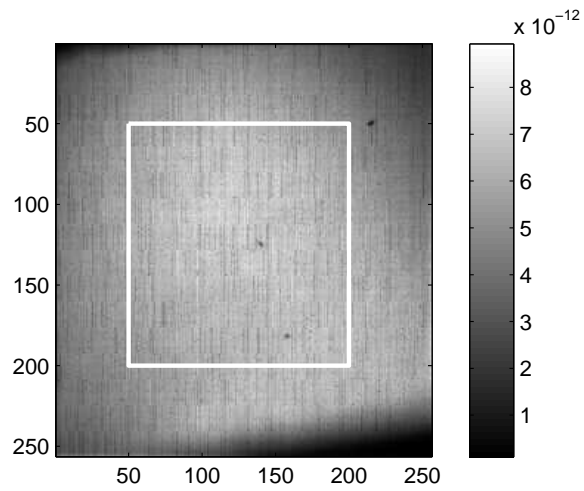


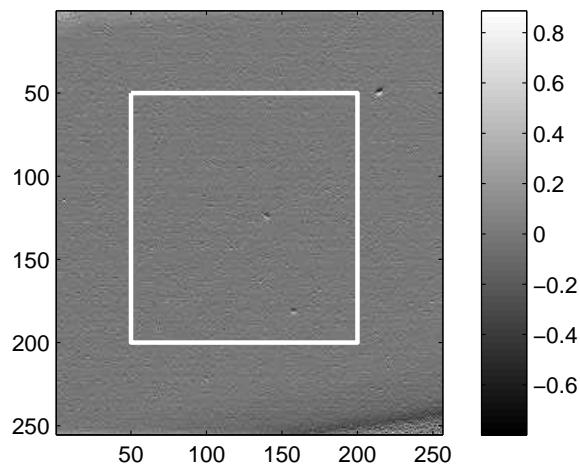
Figure 7.8. Overlapping linear ranges. Since multiple pixels are used at once, input voltages must fall within the linear range of all pixels used. Voltage offsets reduce the overlapping linear range available.

slightly offset from zero resulted. The main concern arising from these voltage offset measurements is the effect on the common linear range of operation along a row of pixels. Since a voltage input is applied along a row, it must be in the linear range of every pixel being used at once on that row. Figure 7.8 shows how two pixels with individual voltage offsets have a reduced overlapping linear input range. If a voltage offset becomes too large compared to the linear ranges of the pixels, Figure 7.5, then special treatment may be needed. Since these pixels are outliers in terms of behavior, they do not necessarily represent an unrecoverable source of error. Schemes for adjusting voltage inputs to take full advantage of the voltage range of the pixels in use at a given time may help. If certain outlying pixels don't allow use with the other pixels then some adjustments to peripheral circuitry could be used to read from them individually.

The resolution of the pixel-plane can also be extracted from an image of a uniform background. Figure 7.9 shows a capture a plain, white background. This was taken using a later pixel plane design as shown in Figure 5.4, which includes extra selection switches in each block. There are still some per-block calibration errors in this capture and a non uniform illumination of the background. To extract the characteristics of the sensors, a first-order difference was taken along the columns of the image and the standard deviation was compared to the mean of the original image to obtain a conservative $4^{1/2}$ bits of resolution.



(a) Illuminated White Background



(b) Adjacent pixel ratios along columns

Figure 7.9. Adjacent pixel mismatch. Taking an illuminated background and differentiating between pixels along columns to find differences between adjacent pixels. At least $4\frac{1}{2}$ bits mantissa SNR between adjacent pixels.

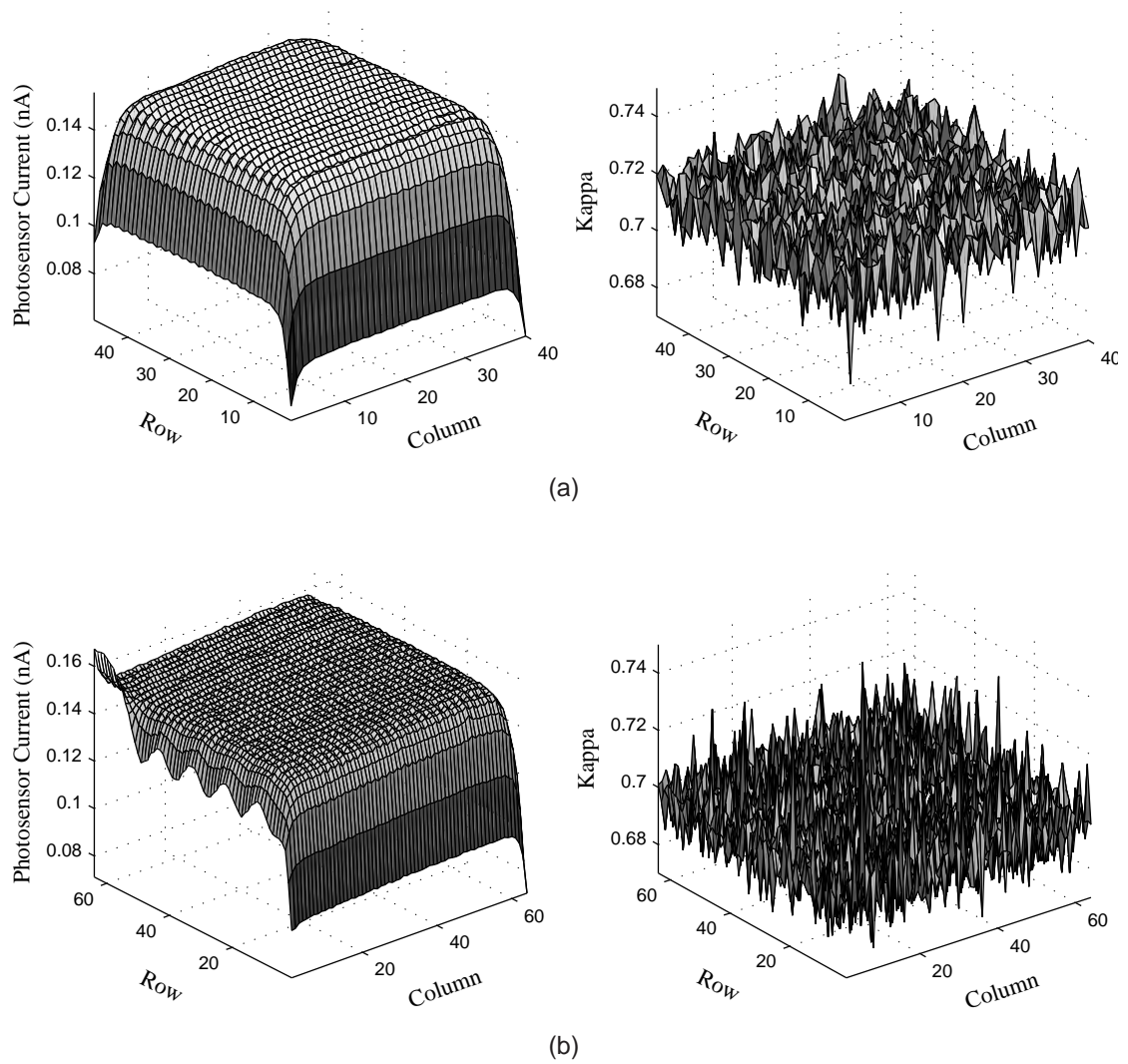


Figure 7.10. Edge effects of two different imager layouts with the same pixel design but different peripheral circuitry. Photosensor current shows variation though kappa does not, meaning the transistors are unlikely the cause of edge effects.

7.2 Pixel Plane Design for Reduced Parasitics

Parasitics in large systems make low-current-based signal transmission difficult. Such a system must mitigate the effects of parasitic capacitances and leakage currents. The capacitances arise from the metal lines running the length of the image sensing plane, the parasitic P-N junctions from all the transistors connected to the line and, bulk and gate capacitances in the channels of any switches that the currents run through. In the current versions of the IC, switches were added to groups of pixels, isolating the parasitics of pixels that are not being accessed.

Pixels along a given row of the image plane share a single differential voltage input, which sets the multiplication factor for the row. Pixels along a column share an output line, utilizing KCL to perform current summation.

Pixels are grouped into 8-pixel tiles with a special set of switches, Figure 7.11. The switches selectively allow the pixels in the tile to output to the column. When deselected, the pixels currents are switched off of the column's output line to a separate fixed potential. Since only a sub portion of the rows of the imager are read at a time, these switches reduce the parasitic capacitance introduced by the deactivated pixels' drain junctions from the 8 in previous design to 1. Furthermore, these parasitic junctions introduce unwanted currents to the output line, since they themselves are photo diodes. This switch therefore reduces parasitic capacitances and currents.

7.3 Offset Removal

Column offsets, as in many imagers, were been found to be the primary source of error in the system. In previous work, the offsets were removed in post processing by removing column averages. However, we now understand the unique nature of the column offsets in this system, allowing us to properly remove offsets and pursue the integration of on-chip correction circuitry.

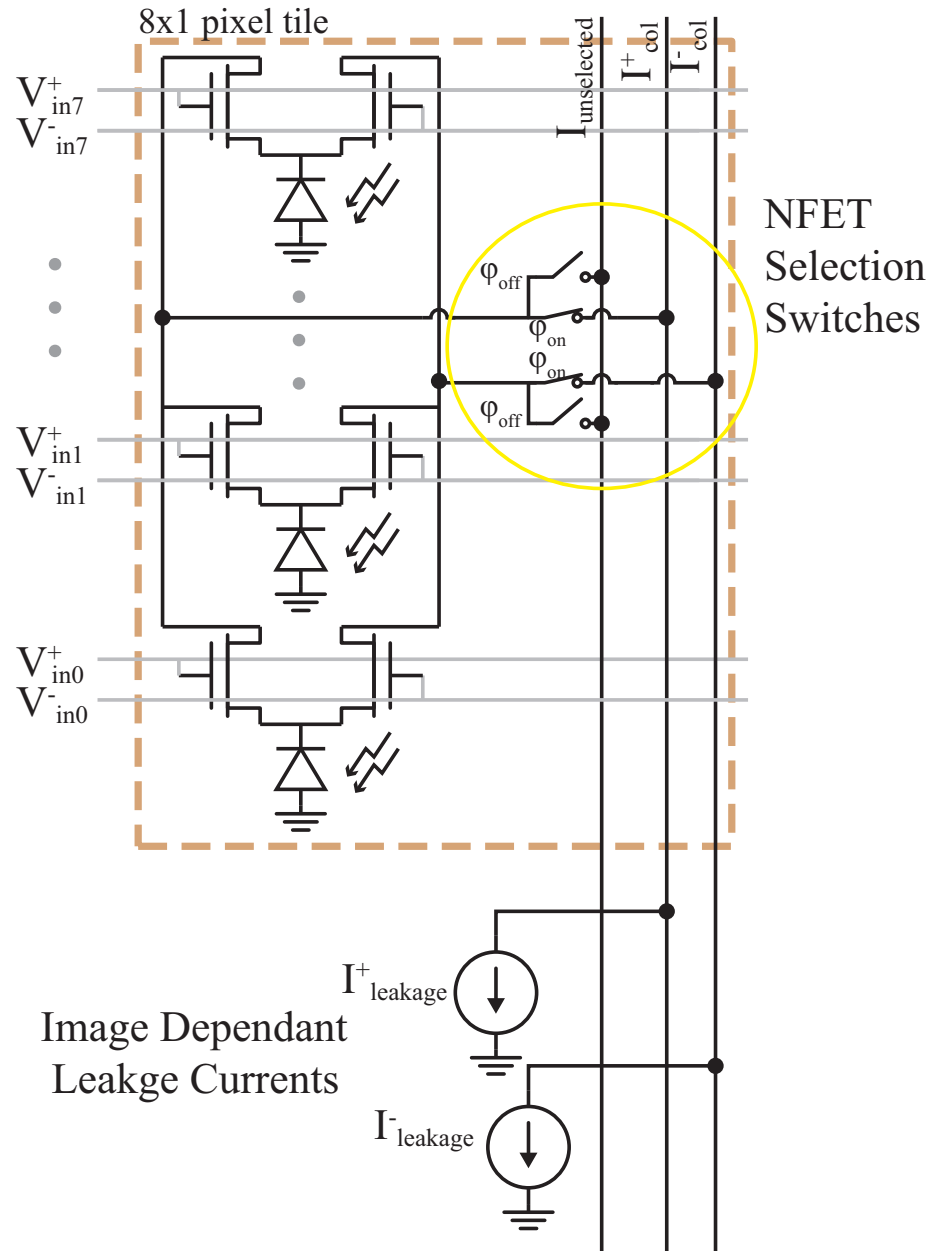


Figure 7.11. Pixels with leakage currents. Switches are introduced in the pixel plane to reduce total parasitic currents and parasitic capacitances on the readout line.

Learning to maximize the utilization of any system, which includes removing or compensating for errors, requires an understanding of often ignore non-idealities. In the pixel plane there is a massive collection of parallel, multiply-accumulate cells. Unfortunately, each device varies, introducing undesirable errors. In particular, we shall first examining the error originating from the differential pair offsets. The offsets are primarily due to the threshold mismatches between the pairs of differential transistors. As can be seen the offsets do not show any obvious spacial correlations that could be easily compensated. The layout of the pixels was done so that possible variations would produce 2-D separable error characteristics that could be compensated.

The error contribution of each pixel due to voltage offsets is a differential error of $V_{offset} \times G_0 \times I_{photo}$ where G_0 is multiplicative factor due to quantum efficiency and kappa variation at a particular pixel, V_{offset} is the offset of the pixel's differential pair, and I_{photo} is the ideal conversion of the light level at that pixel. Of importance here is that the error is a function of I_{photo} , or rather the light level at each pixel. Assuming the image is not static, this error becomes a temporally varying signal which must be uniquely compensated in each frame. This is done by modulating the desired signal component, Figure 7.12. Using positive transform coefficients first and then and negative ones, combined with double sampling, yields two values that when subtracted are free of this error component. On this IC, digital modulation has been added in the Row Control and Readout Control. Since the signals are differential going into and out of the pixel plane, modulation simply involves swapping the positive and negative channels. Figures 7.13 and 7.18 show the results of this operation. This also removes the effects of currents from parasitic junctions which is also image dependent. This procedure works in the case of any transform, not just the identity as shown in the figure. Complete on-chip solutions have been explored and include on-chip integration of the signal modulation-demodulation.

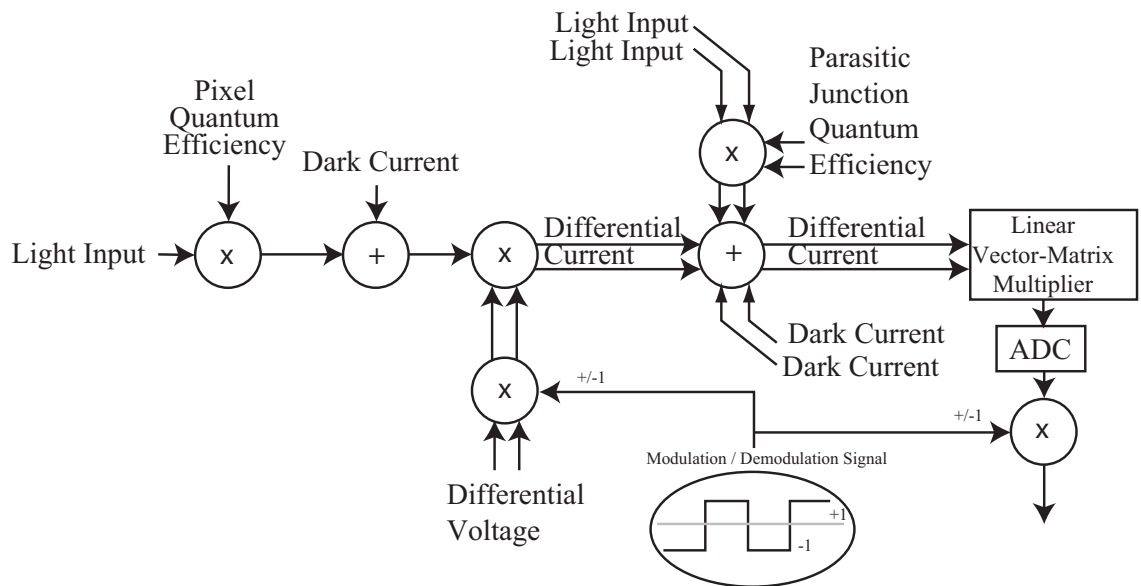


Figure 7.12. Mismatch and parasitic current removal using chopper stabilization

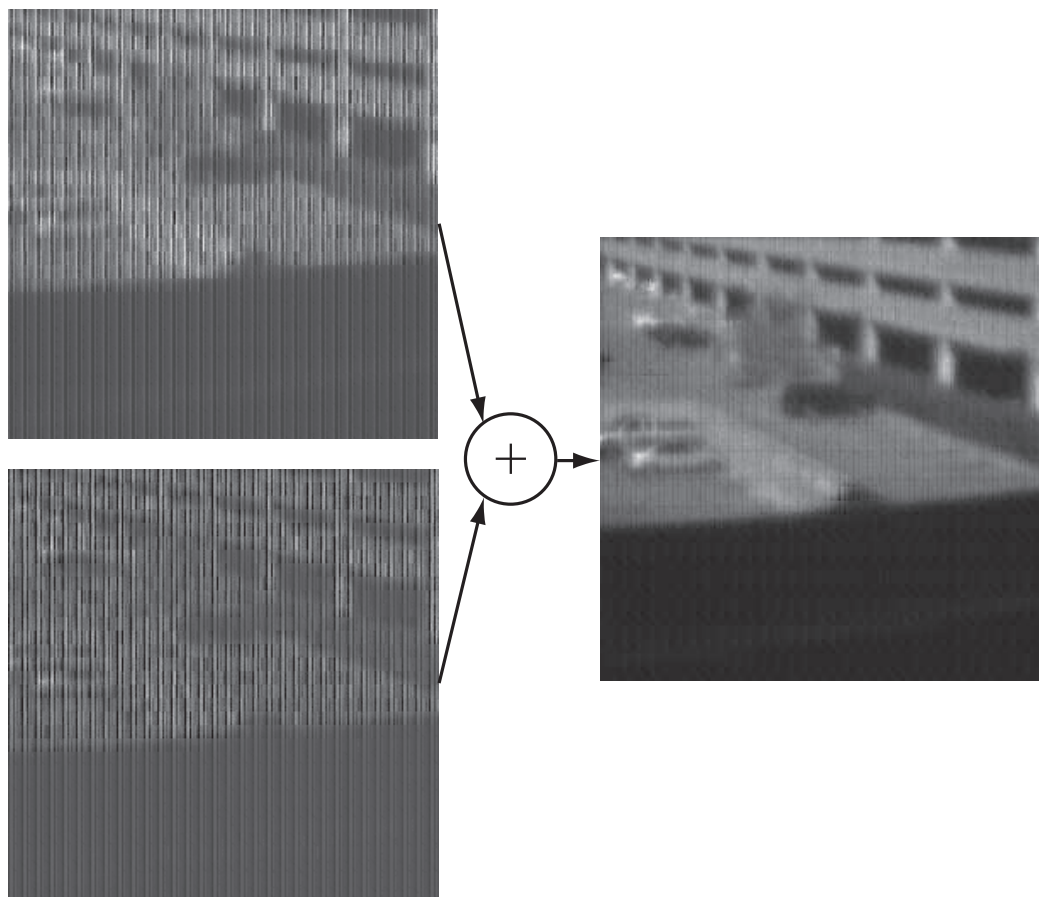


Figure 7.13. Images of mismatch removal on 256x256 imager.

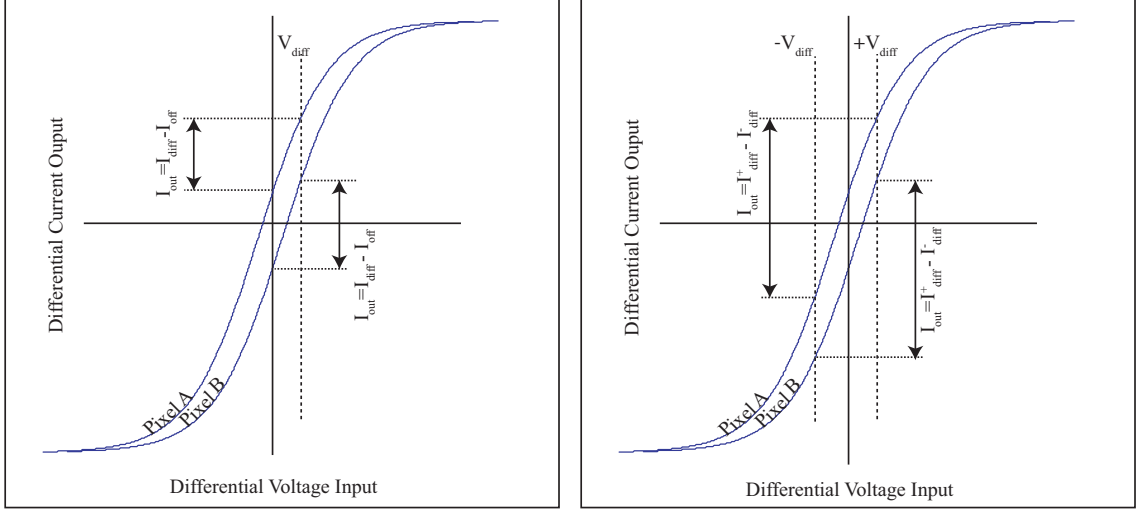


Figure 7.14. Double reading. The subtraction of two reads rejects offsets. The two curves in each (a) and (b) represent the transfer characteristic of two pixels under the same illumination but with different voltage offsets. (a) illustrates current differences taken applying differential voltages of zero differential and V_{diff} differential. (b) illustrates current differences taken applying differential voltages of V_{diff} and $-V_{diff}$.

7.4 Double Reading

To remove the column offsets, we took a difference of two readings while effectively modulating the signal of interest in the presence of the undesired signal component. Figure 7.14(a) shows curves from two pixels under the same illumination. The pixels, or rather the differential transistor pairs in them, have different offset voltages and thus different current offsets. When V_{diff} is applied to the differential inputs of each pixel, the desired output components coexists with the undesired component from the offsets. However, when a reading is taken with $V_{diff} = 0$ and subtracted, the offset error is removed. As an alternative approach, measurements were taken using V_{diff} and $-V_{diff}$ as Figure 7.14(b) illustrates. Creating $-V_{diff}$ turns out to be conveniently implementable as a swapping of the positive and negative differential inputs.

We used these double-read methods with an array to read images. The pixel rows were grouped into blocks of 8. To read an image, the columns of an identity matrix are applied to the on pixels, the ones in the currently selected blocks. The pixels receive differential input

voltages which have a common mode V_{com} . The coefficients are conveyed in the difference of the voltages. The off pixels are those in the unselected blocks. Typically the all the off pixels have there differential inputs all tied to one common voltage referred to as V_{off} . V_{off} may be set as V_{com} for speed reasons or set to ground to reduce the contribution of all the pixels in a column that are not being read. When trying to obtain a direct readout of the image and not a transformed image, an identity transform would be used. An identity transform is a special case where only pixel in a row is read at a time. So the zeros in the identity matrix could be set as either V_{com} or V_{off} . The general case for the transforms is that all the coefficients including zero are generated using a common mode V_{com} . For double reading, two matrices would be applied and the results would be subtracted.

The example of using an identity matrix to read the image will be given here. For the technique illustrated in Figure 7.14(a), first a zero matrix, Equation 7.3, would be used to read the offsets, and then an appropriately scaled identity matrix, Equation 7.4, would be used to read the image. The results from the zero matrix read would then be subtracted from the image read. M_{zero} has the property that all its column vectors are the same, so only one read must be performed for this matrix. The technique illustrated in Figure 7.14(b) involves reading one image using an identity matrix, Equation 7.4, and then a differentially negative version of the identity matrix, Equation 7.5. These are then subtracted to get the final result.

$$M_{zero} = \begin{bmatrix} V_{com} & V_{com} & \cdots & V_{com} \\ V_{com} & V_{com} & \cdots & \\ \vdots & \vdots & \ddots & \\ V_{com} & & & V_{com} \end{bmatrix} \quad (7.3)$$

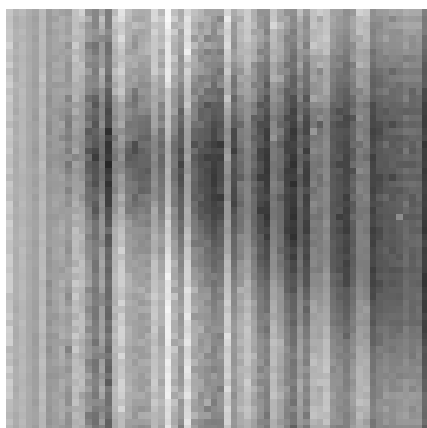
$$M_{plus} = M_{zero} + (V_{diff}/2) I \quad (7.4)$$

$$M_{minus} = M_{zero} - (V_{diff}/2) I \quad (7.5)$$

To aid in subtraction, the negation one of the results can be obtained by switching the differential outputs of the imager. Figure 7.19 shows the architecture of a fabricated chip used to implement the removal of these offsets. For this chip, the final difference of the differential channels is computed off-chip using a subtraction amplifier circuit.

Figure 7.15 shows some of the first results from reading an image. A picture of a cardboard in a roughly triangular shape was imaged in the foreground against the bright ceiling in the background. It may be important to note that the image was not in good focus, so the blurry image is not a result of the imager. The triangular shape was used to illustrate an important point in removing column offsets. Figure 7.15(a) shows a standard image read using a full rail difference on differential pair, approximately 3.3V and 0V, with $V_{off} = 0V$. Figure 7.15(b) shows the same image read with the differential voltages and currents switched in their polarity. The voltages are flipped on-chip using switches placed just before the pixel array. The currents are flipped just after the pixel array. Switching the currents produces a negative result so that adding the two results becomes an addition. The expected column offsets are clearly visible in both of the images. Comparing Figure 7.15(a) and Figure 7.15(b) reveals that flipping both voltage and current negates the column offsets while maintaining the polarity of the image. The image is effectively negated twice while column offsets are negated once. Figure 7.15(d) shows a result much more representative of what the image should be. It is created by adding the results of Figure 7.15(a) and Figure 7.15(b), which have opposite offsets but the same underlying image. These results confirm the expected behavior of the imager array and its offsets.

Figure 7.15(c) shows the results of an attempt to remove the offsets of the image in Figure 7.15(a) by removing the average of each column. This attempt initially may seem reasonable since the column offsets is almost a constant along a column and acts similar to a DC offset. However, doing this removes the DC of the transformed image, which is usually undesirable. The triangular shape helps to emphasize this effect since the resulting image should not have the same average or DC for each column. The leftmost columns



Standard image read
(a)

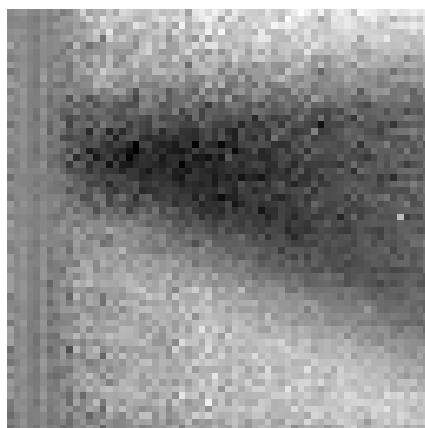


Image from (a) with column average removed
(c)

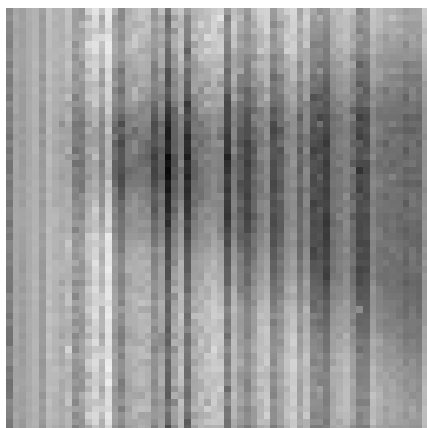
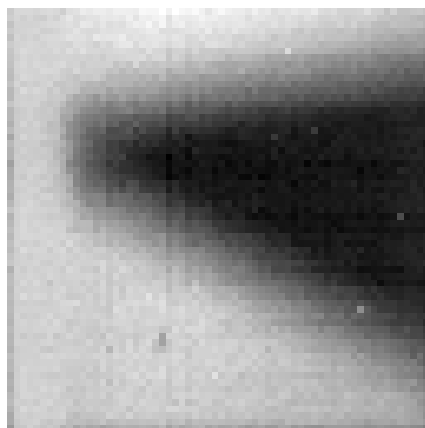


Image read with flipped voltage and current
(b)



Addition of (a) and (b)
(d)

Figure 7.15. Results while reading a raw image. (a) A standard positive read showing column offsets. This is done outside the linear range of the diff pair. (b) The same image with input voltages flipped and output currents flipped. The image maintains its polarity while the offsets are negated. (c) This is an attempt to remove offsets using column-wise mean removal, but it also removes the column-wise means of the desired image. False darkening on the left and brightening on the right occurs. (d) The addition of a and b to remove offsets without removing the desired column-wise means of the actual image

should have the lightest column averages but they were darkened by the DC offset removal technique. The rightmost columns should have the darkest averages and instead are artificially lightened. As Figure 7.15(d) shows, the double sampling technique does not suffer from this problem.

Figure 7.16 shows results of working in the linear region. Figure 7.16(a) is the normal read using an identity matrix scaled to be in the linear region of operation. Figure 7.16(b) shows a read with differential input voltages switched and differential output currents switched. Again, in (b) the image maintains its polarity and the offsets are negated. But, there is an additional anomaly on the right side of the images that shows up as a bright area in the image. A read using a zero matrix (c) shows the same anomaly. Adding the results of the positive and negative reads cancels most of the offsets but the anomaly remained, Figure 7.16(d). Using the zero matrix to remove offsets produced very good results, Figure 7.16(e). The results in (d) are better except in the region of the anomaly. The anomaly and the artificial edges near it are like due to a nonlinearity problem with the I-V converters on the chip when currents are low. Since the right hand side of the image has the lowest currents it became a problem though. Figure 7.17 shows results of a DCT transform and using the zero matrix to remove the offsets. Once the linearity problem is corrected, using positive and positive reads may produce even better results.

Figure 7.18 shows the double sampling technique used on the 256x256 imager. The large column striations can be seen in the two raw images, but when added the offsets are gone.

7.5 Dual-Slope Integration

The ability of the chip in Figure 7.19 to reverse the polarity of the output was originally conceived to allow on-chip offset removal. Reversing the polarity of the outputs implements a negation and a temporal integration implements a summation. So, this chip can implement the subtraction of two results on-chip. The outputs of the two integrators are

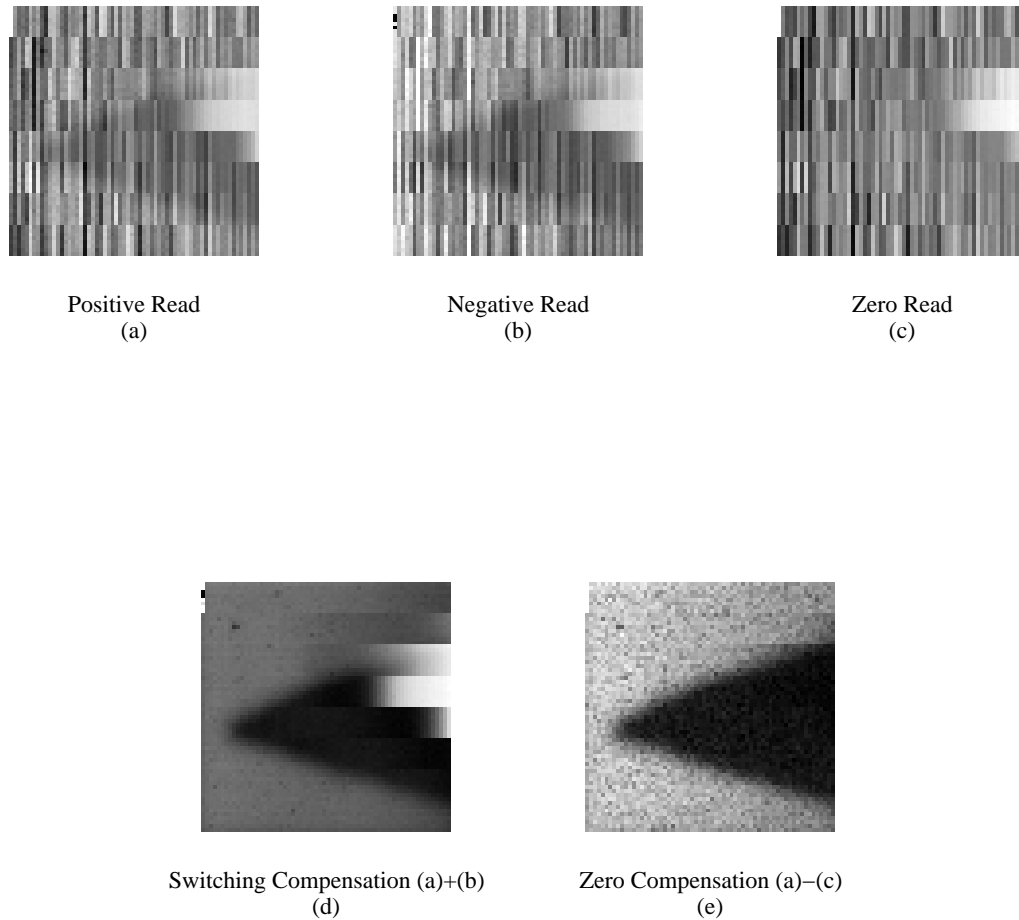
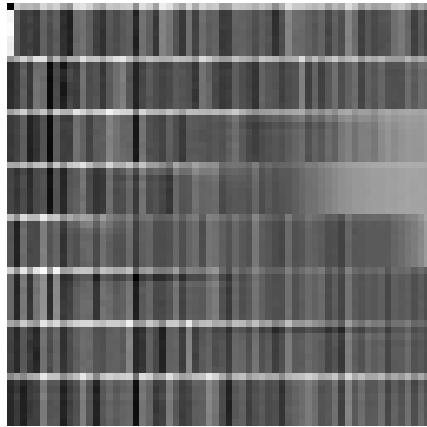
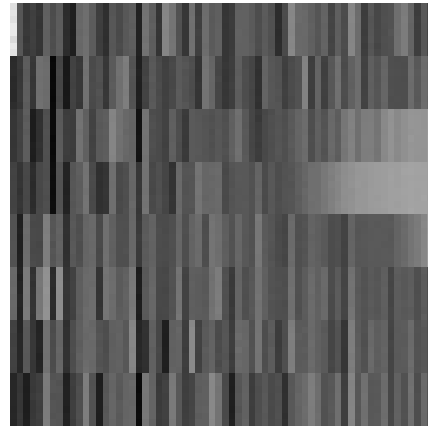


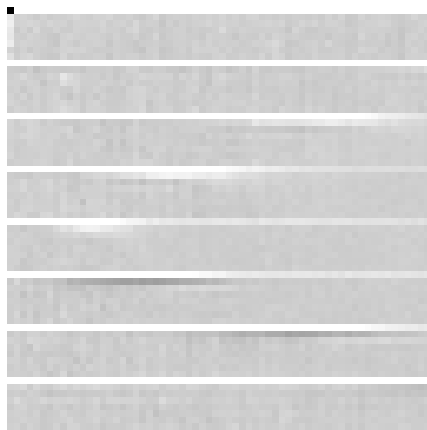
Figure 7.16. Results while reading an image using an identity matrix transform in the linear region with “off” blocks set to 0 V common mode. (a) shows an image read using the identity matrix and (b) shows the results using a negative identity matrix and negated outputs. (c) shows a read using a matrix of all zeros (1.5V common mode). (d) shows the result of the addition of (a) and (b). The white anomaly on the right hand side is likely a result of the I-V converter’s nonlinear response which can be fixed in a future design. (e) shows zero matrix correction using (a)-(c). This avoided the white artifact but, as in (d), some false edges occur at the block boundaries, also likely due to the nonlinearity of the I-V converters.



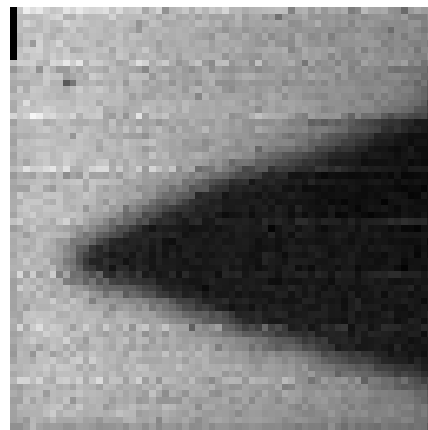
Raw DCT 1-D
(a)



Zero Matrix Read
(b)



DCT - Zero Matrix
(c)



Reconstructed Image from (c)
(d)

Figure 7.17. DCT offset removal results using a zero matrix read. (a) shows a 1-D DCT computation and (b) shows offsets read using a zero matrix. (c) shows the transform with the offsets removed and (d) shows the result of performing an inverse DCT on (c).

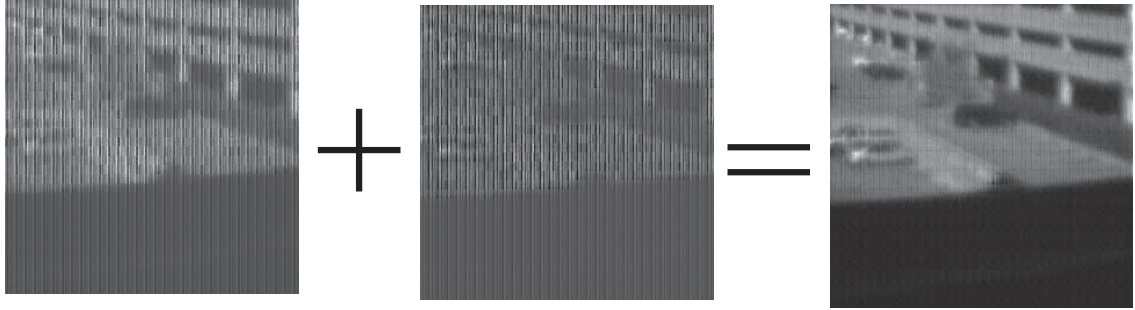


Figure 7.18. Mismatch removal on 256x256 imager. The first image is a read of the data under an identity transform. The second image has the inputs and outputs of the pixel plane reversed such that the image maintains polarity while the block-wise and full-frame column offsets are reversed. These column offsets are caused by voltage offsets in the pixels' differential pairs and parasitic diode junctions which conduct current based on the light falling on them. The addition of the two raw transformed images results in the removal of the errors from the voltage mismatches and the parasitic junction currents.

shown in Figure 7.20 along with an amplified and offset subtraction of the two. To begin, the appropriate row of the input voltage matrix is applied to the imager. The reset of the integrators is released to begin integration and this continues for some time. Then, while still integrating, the input voltages and output currents are reversed in their polarity. After the negative integration time equals the positive integration time, the outputs are sampled. In this way, the results of the positive and negative versions of the input are created and subtracted temporally on-chip.

Though it is difficult to see, the slopes of the differential outputs change slightly from before to after the polarity is flipped. This is because the desired signal is riding on a large common mode current. The large current offsets complicate the offset removal. The feed-through effects of the switches can be seen at the polarity switching time. Since these effects are proportional to the large offset component of the output, the errors can be large compared to the desired signal. Circuits that reduce signal dependent feedthrough are, therefore, critical for this technique. There are also some nonlinear effects in the amplifier used in the integrator. The initial curvature does not actually affect the final result as long as the integrators reach a linear region before they are read.

Figure 7.21 shows a comparison of results taken from the imager. Figure 7.21(a) shows

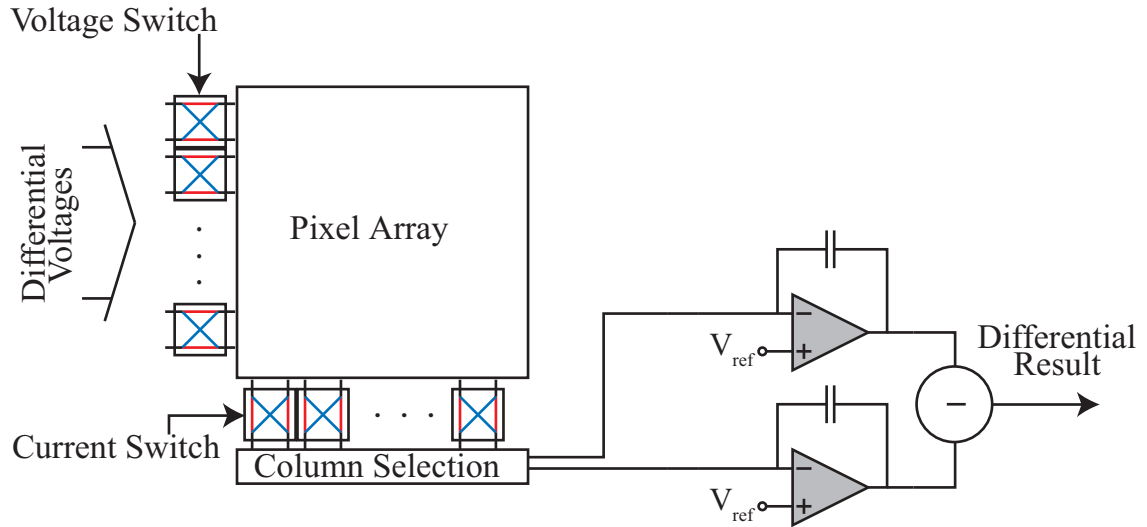


Figure 7.19. Switch imager design for double reading and dual slope integration.

results using the dual- integration method discussed here while Figure 7.21(b) shows results taken using two separate integration cycles. Though the dual integration removes most of the current offsets, it seems that double sampling with two separate integration cycles produced better results. This should be expected since nearly all offsets are produced identically in the two integration cycles and thus would be canceled. Further circuit design including efforts to improve the linearity of the output amplifiers and reduce feedthrough effects may narrow this margin.

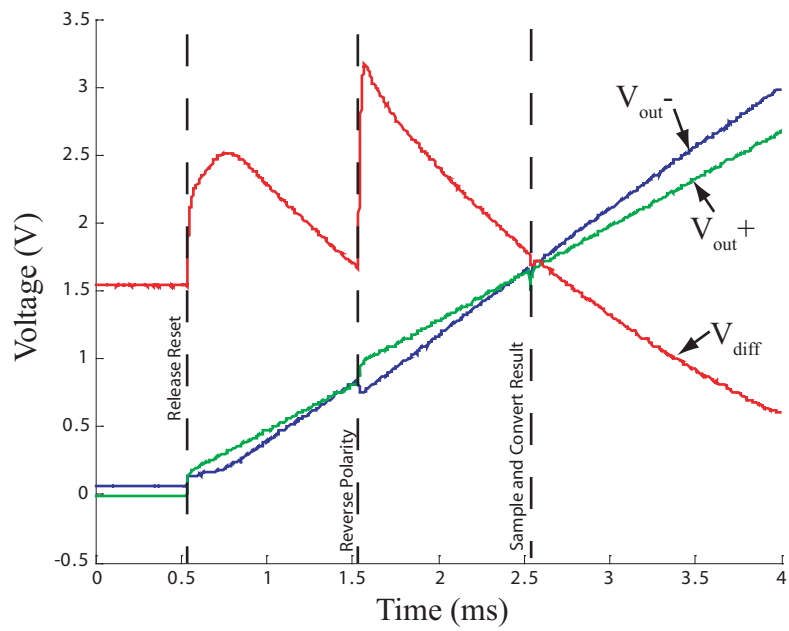
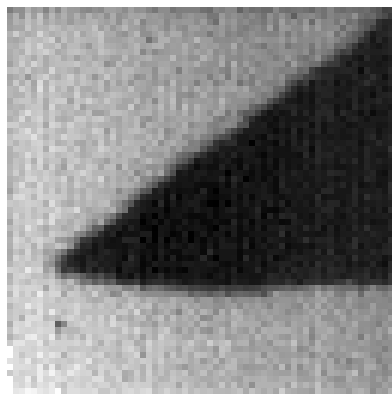
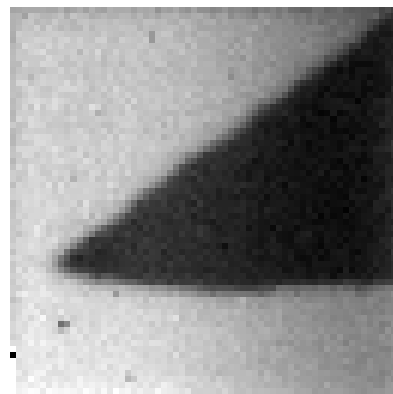


Figure 7.20. Dual slope integration voltage outputs.



(a) Dual-Slope Integration



(a) Double Sampling

Figure 7.21. Dual slope integration vs. double reading results.

CHAPTER 8

APPLICATION IN COMPRESSIVE SENSING

The standard model for sensing and sampling information includes the requirement of sampling at the Nyquist rate. This is necessary to uniquely convey all the information in the signal being sensed. Often, preexisting knowledge can reduce the amount of data required to uniquely capture the information in the signal. But, without a mechanism to capitalize on a priori knowledge in the sensing process, the sensor and communication hardware must exhaustively sense, process, and transmit information at the Nyquist rate. A compression stage can ease the throughput requirements of communication channels, which is especially critical for wireless sensors, but the advantages are only seen by the stages that follow the compression stage. These advantages translate to lower power consumption and smaller sizes.

More significant reductions in power and hardware complexity can be achieved if data reduction is performed earlier in the sensing chain, Figure 8.1. The reductions are a result

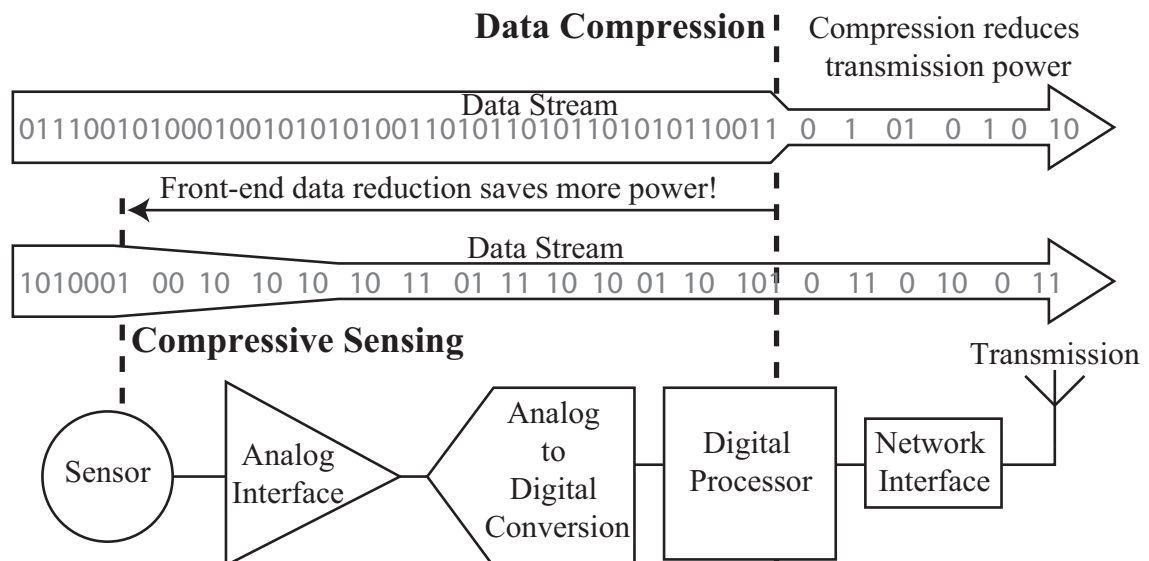


Figure 8.1. Compressive Sensing system design. Total data manipulation and power is reduced in the chain from sensor to transmitter by sampling less often instead of just compressing data in the digital domain.

of reducing the data throughput across more stages in the sensing system. In the extreme case, where data reduction is done at the front end of the system, all stages receive these benefits. This translates to less total system communication and possibly less computation required at the sensing device. Offloading computational complexity, like decoding, to the receiver is often more efficient since the receiving system often has relaxed power and area constraints, as is the case with distributed wireless sensor networks utilizing a central processing node.

Front end data reduction is exactly what compressive sensing enables [31–34]. Compressive sensing exploits the knowledge that the signal or image being acquired is *sparse* in a known transform domain (e.g. the wavelet domain). In other words, there are fewer degrees of freedom in the signal than the Nyquist rate requirement implies, so fewer samples are needed to capture the signal. Presently, in the majority of vision systems, the data throughput required through most of the system is much larger than entropy rate of the signals being processed. This suggests that fewer bits could be used to represent the signal in the system. As a result, compressive sensing is particularly well-suited for image sensing applications, and the development of hardware well-suited to compressive sensing is critical to realizing the anticipated power and size savings or increased performance, such as the single-pixel camera discussed in [35].

While several technology options exist for image sensing applications, CMOS-based image sensors, also called imagers, share essentially the same manufacturing processes as those used for standard VLSI implementations. Complex computational circuitry can therefore be combined with the sensors and interface circuitry. This chapter discusses the capability of a computational image sensor to implement compressive sensing operations. The structure implements a computational architecture similar to that in [6]. The current image sensor design was implemented on a 22.75 mm² die in a standard .35 μm CMOS process. The resolution is 256 \times 256 with a pixel size of 8 μm \times 8 μm .

The fundamental capability of this image sensor can be described as a matrix transform: $Y_\sigma = A^T P_\sigma B$, where A and B are transformation matrices, Y is the output, P is the image, and the subscript σ denotes the selected 16×16 pixel sub-region of the image under transform. This separable transform operation is demonstrated in hardware to be sufficient to perform compressive sensing.

8.1 Transform Image Sensor

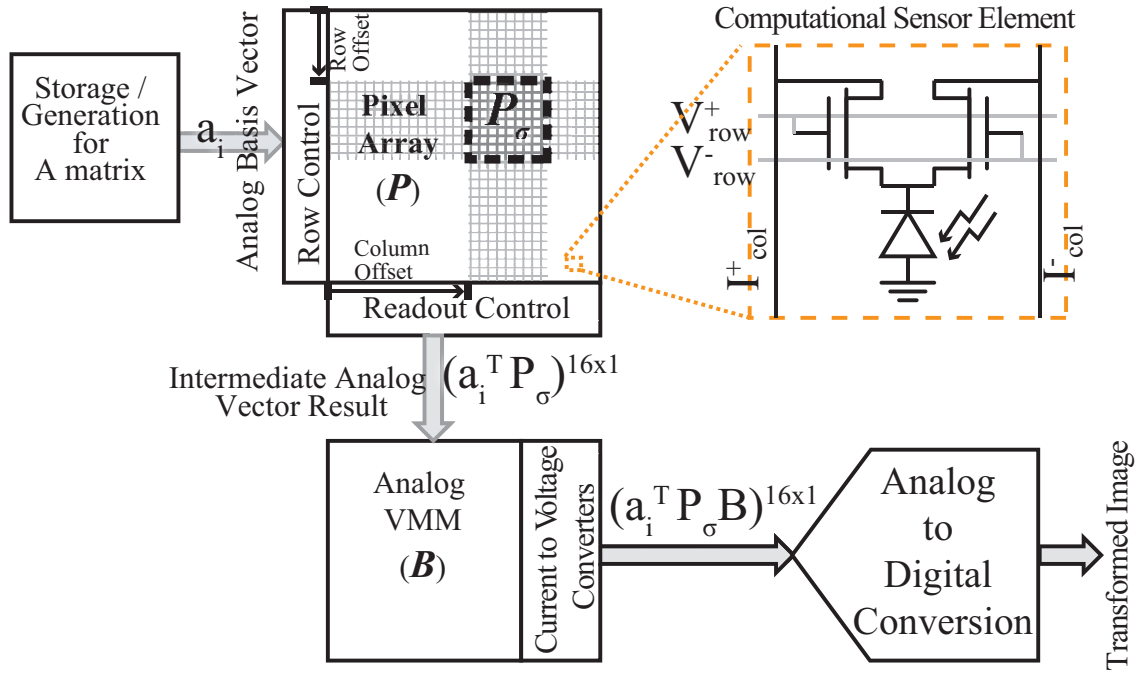


Figure 8.2. Separable transform image sensor hardware platform with the capability to capture reduced data sets through projections onto reconfigurable sets of basis functions.

The separable transform image sensor uses a combination of focal-plane processing performed directly in the pixel, and an on-die, analog, computational block to perform computation before the analog-to-digital conversion occurs.

The first computation is performed at the focal plane, in the pixels, using a computational sensor element shown in Figure 8.2. It uses a differential transistor pair to create a differential current output that is proportional to a multiplication of the amount of light falling on the photodiode and the differential voltage input. This operation is represented

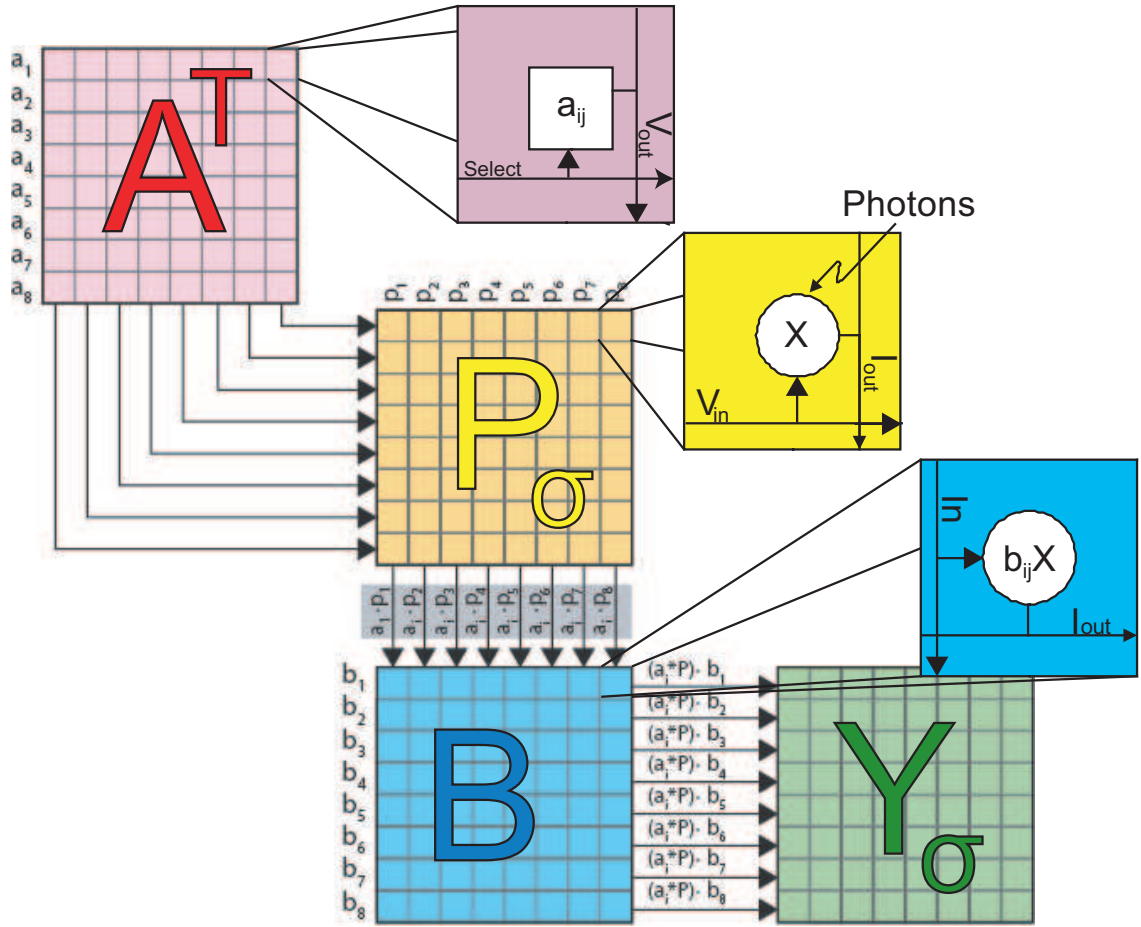


Figure 8.3. Block matrix computation performed in the analog domain. Illustrated here as an 8×8 block transform, both a computational pixel array and an analog vector-matrix multiplier are used to perform signal projection before data is converted into the digital domain.

in Figure 8.3 as the element for the P_σ block. The electrical current outputs from pixels in a column add together, obeying Kirchhoff's current law. This aggregation results in a weighted summation of the pixels in a column, with the weights being set by the voltages entered into the left of the array. With a given set of voltage inputs from a selected row of A , every column of the computational pixel array computes its weighted summation in parallel. This parallel computation is of key importance, reducing the speed requirements of the individual computational elements.

The second computation is performed in an analog vector-matrix multiplier (VMM) [4]. This VMM may be designed so that it accepts input from all of the columns of the pixel array, or it can be designed with multiplexing circuitry to only accept a time-multiplexed subset of the columns. This decision sets the support region for the computation. The implementation used for these experiments uses the time-multiplexed column option. The elements of the VMM use analog floating-gate transistors to perform multiplication in the analog domain. Each element takes the input from its column and multiplies it by a unique, reprogrammable coefficient. The result is an electrical current that is contributed to a shared row output. Using the same automatic current summation as the P matrix, a parallel set of weighted summations occur, resulting in the second matrix operation.

8.2 Sensing with Decorrelated Basis Functions

A common mathematical scenario entails a signal whose energy is spread among many basis functions in one domain, and only a few in another domain. The goal of compression can be simplified as the intent to represent as much of a signal's energy as possible with as few coefficients as possible. The choice of the basis functions is normally key to compression performance. Luckily, experience tells us that for natural images the discrete cosine transform (DCT) basis is a good choice because most of the image energy usually falls into the so-called low frequency components. A large number of low-valued, high-frequency components can be neglected at the cost of losing some edge fidelity. Having this a priori

knowledge about exactly which of the basis functions are needed to represent the signal enables transmission of the fewest coefficients with minimized overhead.

The problem is that the signal energy is not always what is most important to capture, particularly in images where edges are the most important to be maintained. Wavelets have proven to be a better compression basis, in general and especially for maintaining edge fidelity. The remaining challenge is that even though there are fewer coefficients needed, there is a lack of a priori knowledge about exactly which basis functions are needed.

The scenario of not knowing the optimal subset of basis functions usually means that a complete set needs to be acquired before it can be pruned down. Work in the field of Compressive Sensing [31–34] suggests an alternative, non-adaptive approach, where a seemingly random set of basis functions can be used to sense and transmit data. The basis functions are not prescribed to be correlated with the data, eliminating the problem of choosing an optimal set of observation functions. Instead, the optimization burden is shifted to the receiver, which finds an optimal estimation based on a cost function rewarding sparsity in a chosen domain and consistency with the observations. So, the a priori knowledge is not embedded in the sensing or transmitting functions, but instead in the signal reconstruction process. It should be noted that since the observation functions are not correlated to the data or the reconstruction basis, each one statistically has about the same signal information content and contributes with equal probability to each of the reconstruction basis functions. In our study, we utilize the noiselet basis functions for our observations [36]. Noiselets are an orthogonal basis of waveforms which, for our intents, behave like random waveforms (see [33] for a more detailed discussion).

8.3 Results

The analog computational system described was used to sense images as projected onto programmed basis sets. The raw pixel-by-pixel data is never transferred through the system. Instead, the two-step computational process at the front end of the system projects the

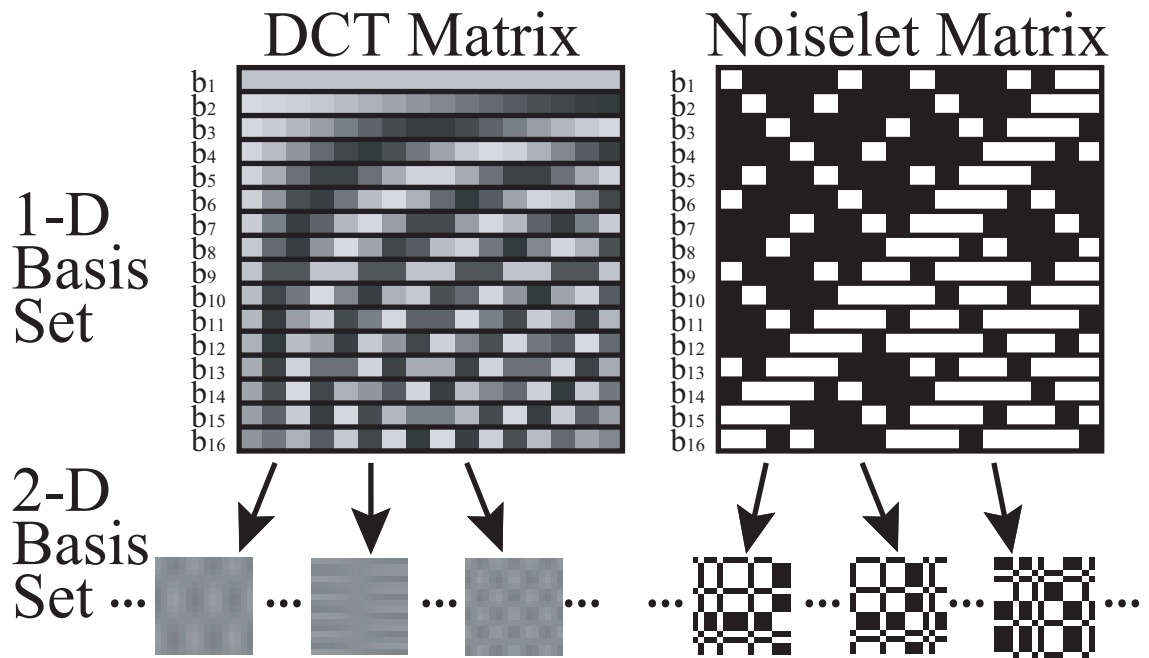


Figure 8.4. DCT and Noiselet basis functions. The DCT 2D basis functions are structured to correlate with different spatial frequencies in images. The inner products with the different DCT basis functions are generally non-uniform, since most of the energy in images lies in the low frequency components. The noiselets basis are decorrelated with most image features and with reconstruction basis functions, making each noiselet basis function statistically as significant as any other.

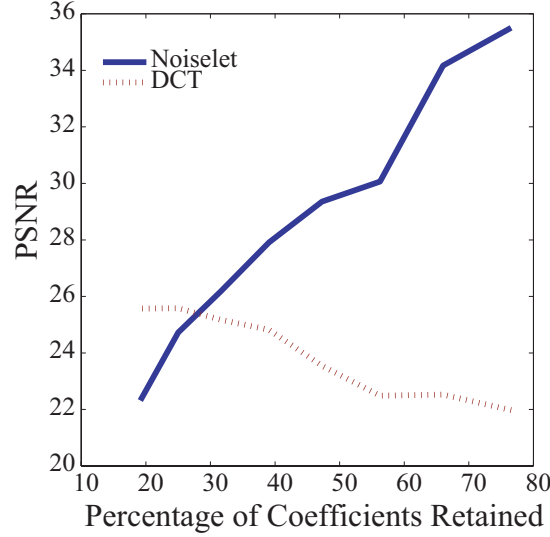


Figure 8.5. PSNR of reconstruction vs. percentage of used transform coefficients. As expected, retaining a small number of DCT coefficients gives better performance than using a similar number of noiselet transform coefficients since the signal is concentrated in the low frequencies. However, as more DCT coefficients are used, the SNR drops because the analog system contributes an equal noise with each additional coefficient but less and less additional signal. When more coefficients are used, the noiselet-based reconstruction performs better. This is likely because the noiselets consist of only -1 and 1 , and thus can be scaled to maximally use the full analog range. The Noiselet-based reconstruction also benefits from a reconstruction algorithm that optimizes over the entire image.

image onto selected basis and outputs the inner products from this process, which will be referred to as the transform coefficients hereafter. The output of the image sensor IC is therefore the representation of the image in the selected vector space. Performing a subset of the complete projections can either reduce power consumption or increase frame-rate.

In the experiments, a complete set of transform coefficients were collected, and the reduced collection was simulated by discarding measured values. The nonlinear recovery algorithm discussed was used to reconstruct the images captured with Noiselet measurement functions. A pseudo-inverse was used to reconstruct images from incomplete DCT measurements. Since the exact original image is not available, reconstructed images corresponding to incomplete collection were compared against denoised versions of images created from complete coefficient collection.

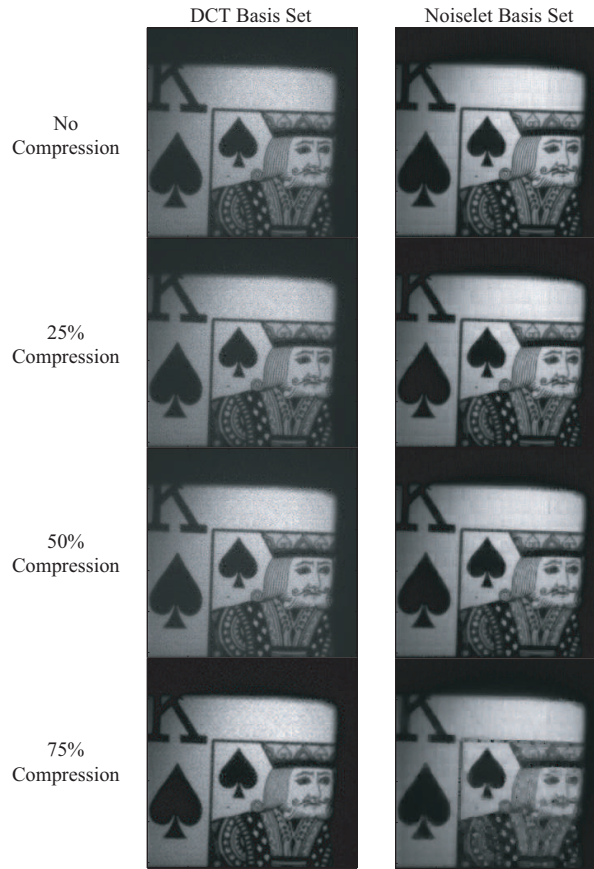


Figure 8.6. Reconstruction results using DCT and noiselet basis sets with various compression levels. The image sensor measured 16×16 blocks of the image projected onto DCT and noiselet basis functions. Subsets of the data were taken and used to reconstruct the shown images using a pseudo-inverse for incomplete DCT measurements and a nonlinear-total-variance-minimization algorithm for the noiselets.

At high levels of compression, retaining few transform coefficients, the DCT representation lead to better peak signal-to-noise ratio (PSNR), Fig.8.5 and Fig.8.6. This is possible because the predefined DCT coefficient removal process exploits the knowledge of where energy compaction occurs in the DCT domain. In the case of the noiselets, higher transform coefficient retention lead to better performance, surpassing the DCT results in quality. It is expected that every transform coefficient in the noiselet domain statistically contributes the same signal and noise power to the resulting image as any other coefficient. In the case of DCT transform coefficients, the coefficients representing high spatial frequencies contribute the same noise as the coefficients representing low frequencies, but they contribute

little signal power. In this case, where the reference images were denoised and have little high frequency information overall, the high frequency components contributed negatively to the SNR. Additionally, the noise in the DCT images is higher than the noiselets because the DCT basis functions are smaller in magnitude than those of the noiselets when implemented in this analog system. The basis functions are constrained to a linear input range of the analog computational elements. Since the noiselet functions consist of only 1's and -1's, they use the fullest signal range of the system, resulting in better signal to noise ratio. Moreover, the noiselet-based reconstruction benefits from a reconstruction algorithm that optimizes even across block boundaries. The analysis of the system behavior is ongoing.

8.4 Conclusion

In this work, we demonstrated a computational sensor IC capable of a unique and flexible set of sampling modes applicable to Compressive Imaging. The capabilities of the IC to reconfigurably sense and processes data in the analog domain provides a versatile platform for compressive sensing operations. To demonstrate the platform, images were sensed through projections onto noiselet basis functions that utilize a binary coefficient set, $\{1, -1\}$, and DCT basis functions that use a range of coefficients. The recent work in the field of Compressive Sensing enabled effective image reconstruction from a subset of the measurements taken. The fundamental architecture is flexible and extensible to adaptive, foveal imaging and adaptive processing in combination with non-adaptive Compressive Sensing.

CHAPTER 9

COMPUTATIONAL RESULTS

Often the appropriate performance metric for a component is not a straight-forward thing to determine. Depending on the end goal of a system and the nature of the environment and the data, several error metrics can be used. What is really important is the final success or failure of the system, but there are complex and often flawed mappings between component error metrics and the system's performance in an application. Even so, quantitative results can at least be useful indicators of system performance and the associated analysis can bring about useful insights to suggest optimal usage and possible design improvements.

To obtain some quantitative results, in addition to the mismatch data presented already, pertaining to the computational performance of the imager, an on-chip two-dimensional DCT calculation is compared to raw identity-transform image capture. Figure 9.1 shows the derivation of the error image and reference image which will be analyzed. First, the imager was placed in a raw access mode to read the pixels values. This resulting raw image serves as a reference for the following DCT computations. The raw image incorporates the mismatch of the pixels and associated interface circuitry. A DCT was performed on-chip looking at the same scene. The DCT results were fed through an ideal inverse DCT on a computer to reconstruct the original sensed image. The reconstructed image was subtracted from the raw image, with some normalization, to give an error image which can be analyzed. The error image represents the errors in the DCT computation. These errors can be extracted and analyzed along with other statistics about the error image. Figure 9.2 shows the data obtained for analysis.

A common picture was presented to the imager, Figure 9.2(a). First the imager was used in an optimal mode to capture the raw image. While this can be fit into the general, computational sensing framework described, $A^T PB$, as capturing the image and processing

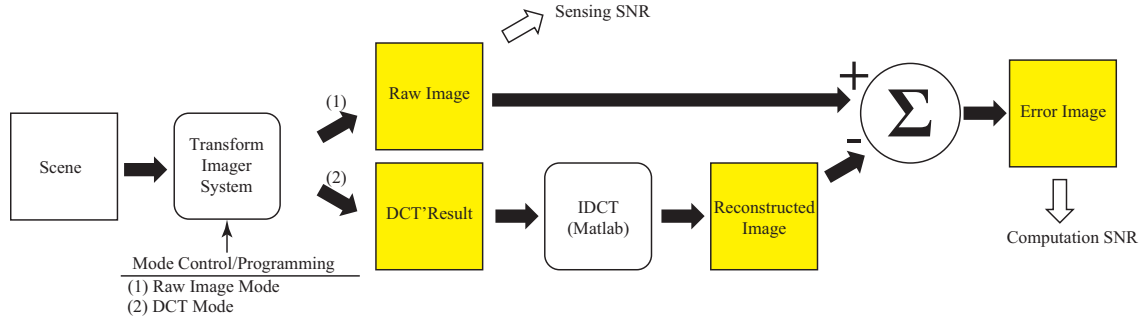


Figure 9.1. System error derivation. (1) The raw image encapsulates errors in sensing and allows analysis of the acquisition quality. (2) The on-chip DCT and ideal IDCT operations create a reconstructed image to represent the analog computational abilities. Subtracting the results provides an error that represents the effect of the analog computations. This error can be used to calculate the effective SNR of the computations.

it with an identity transform, there are some subtle differences when compared to the DCT. The identity matrix is sparse. So, instead of having several elements computing multiplies-by-zero and contributing noise, they can be deactivated. By doing this the reference data is obtained, Figure 9.2(b). The DCT, representing a complex computation, was used to capture the same scene and the result is shown in Figure 9.2(c). With the physical data taken, the results were compared.

To compare the results, two obvious choices are to either perform an ideal DCT on the reference data, Figure 9.2(a), or perform an ideal inverse DCT (IDCT) on the DCT data. The DCT and IDCT both preserve energy, so the latter was chosen for better visual presentation. The IDCT is performed on the DCT data in MATLAB to reconstruct the original image, Figure 9.2(d). The difference between the reconstructed data and the reference data is calculated in MATLAB to produce an error image, Figure 9.2(e). Even in this simple experiment, there were slight image registration issues (slight image shifts) between the data sets that could not be experimentally eliminated in the setup used. Some manual shifting was performed on the data to minimize this.

As a first comparison, the histograms reference data and reconstructed data are shown in Figure 9.3. The image intensities were shifted and scaled to match as best as is possible for comparison. The ratio of standard deviation to mean of each is shown in the figure.

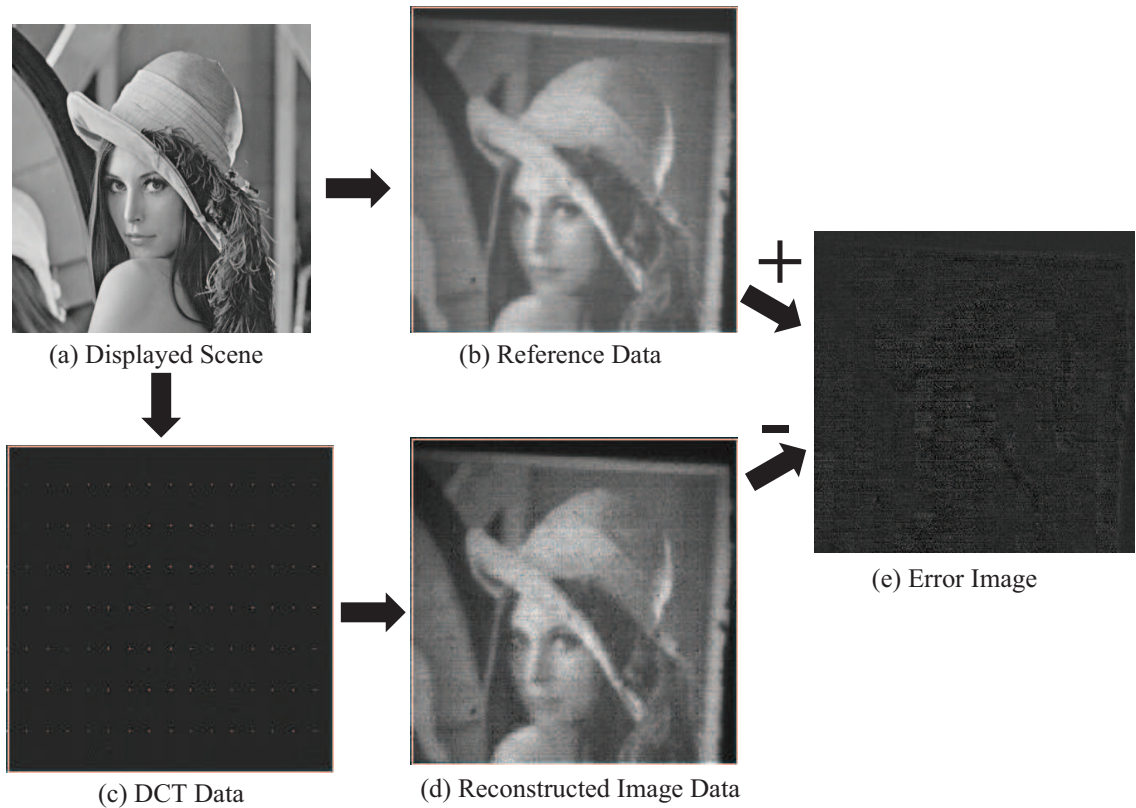


Figure 9.2. Identity data, DCT data, and error image. (a) Image displayed to computational imaging IC. (b) Resulting data from system with imager set to perform an identity transform. This serves as a reference image. (c) Resulting data from system with imager set to perform a DCT. This data is a the representative for complex computations. (d) To compare the DCT results to the reference image, an ideal inverse DCT is done in MATLAB to reconstruct the original image. (e) An error image is produced from the subtraction of the reconstructed data from the reference data.

It can be seen that the noise in the reconstructed data increased the standard deviation. The noise also tends to disperse the pixel values producing an low-pass like version of the reference data histogram.

Keeping the optimal scaling used to match the histograms, the RMS of the error image was compared to the RMS of the reference data. A ratio of 1.14 : 1.00 was found. To determine the nature of the energy, the energy of the error image was removed incrementally by performing a DCT on it and removing components: first the high frequency components and then the low frequency components. Figure 9.4 shows the result. This indicates what the effective results would be if the image data was compressed by using fewer samples.

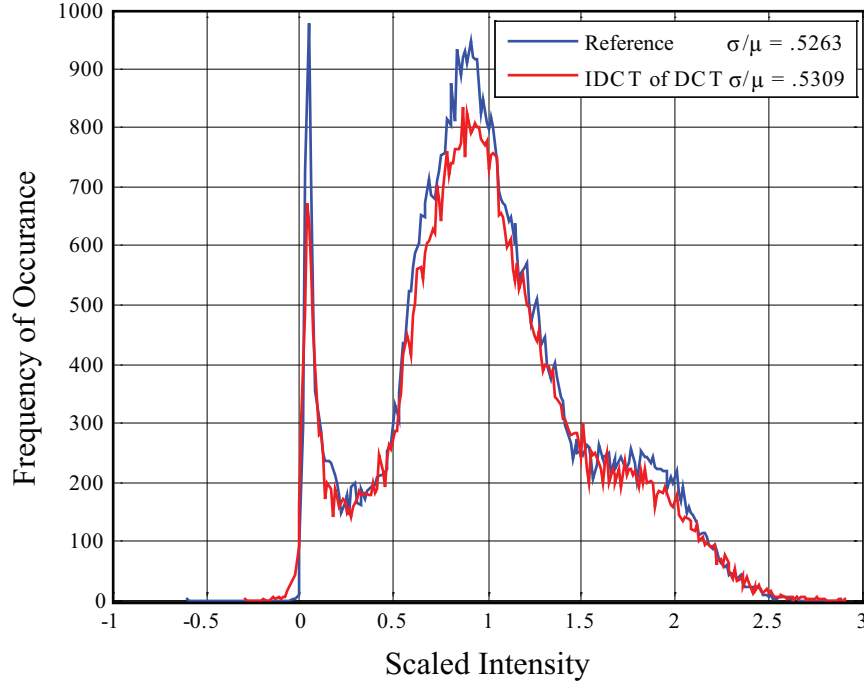


Figure 9.3. Histograms of reference data and reconstructed data.

By the time 30% of the DCT data was eliminated, half of the error image energy was eliminated. This suggests that the computations of the high frequency components are the most flawed.

Further investigation was done into the type of compression this system might perform in order to reduce the number of samples taken. The ideal scenario for compression here would be that fewer basis functions are used to measure the image rather than measuring it completely and performing compression afterward. The incomplete measurement is modeled by removing rows and columns of the DCT block results. A more traditional method would remove diagonals of the results. To see if there was a significant difference between the approaches, they were both calculated from the data taken. Figure 9.5 shows the ratio of energy in the error image removed as a function of the percentage of coefficients removed, both for the case of diagonal coefficient elimination and for row-column coefficient elimination. The results show the approaches produce similar performance, suggesting the limitation of row-column based coefficient removal is not a detrimental one.

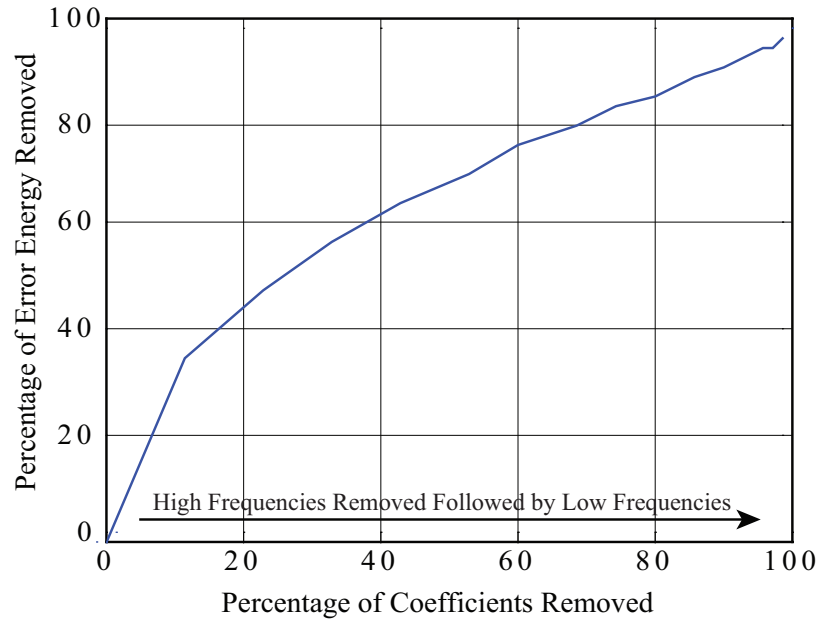


Figure 9.4. Error energy loss in compression. Correlating the error image energy to DCT coefficients can be done by examining the loss of the energy in the error image when DCT coefficients are ignored.

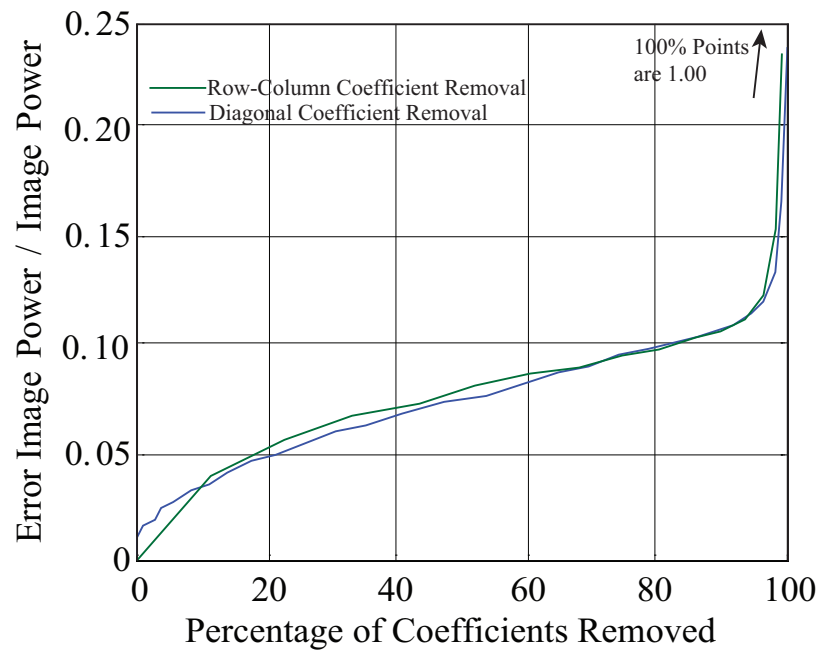


Figure 9.5. Error loss with non-standard compression. In order to perform fewer conversions, our imager design performs the equivalent of removing rows and columns of the output matrices instead of removing diagonals. These approaches are compared here. In one case compression, is performed by removing diagonals of the DCT coefficients, and in the other, rows and columns are removed. In both cases, coefficients representing high spatial frequencies are removed first. The results are similar.

CHAPTER 10

CONCLUSION

This thesis described the development of a computational image sensor capable of performing image processing in the analog domain. Among the primary advantages is the ability to filter data before it undergoes a costly analog-to-digital conversion. The ability to keep the signals in an analog format makes available the ability to utilize other low-power analog processing blocks such as analog-domain classifiers. Analog processing elements and systems can work efficiently in parallel to achieve many of the same calculations achieved in the digital domain, but more compactly and with lower power consumption. Part of this is due to the ability to process parallel with fidelity better suited for processing components of natural signals, and part of this is due to the lessening of costly inter-chip communication. Since the imager has been designed on a standard CMOS process, higher levels of integration are possible. ADC converters and even digital processors can be integrated on the same IC as the analog and sensing circuitry. This can lower the cost of a system while the programmable nature of the IC maintains the flexibility needed for a variety of potential applications. The developments of the components of this imager and the elements of the investigation of those components and the system are summarized in this chapter.

10.1 Detailed Computational Pixel Investigation

My work started with the design, fabrication, and intricate testing of ICs to characterize the computational pixel's operation. Using some ICs designed by Dr. Abhishek Bandyopadhyay and others of my design, the investigation thoroughly verified light-voltage multiplicative operation of pixels by the use of controlled light levels and detailed experimentation. These experiments were performed repeatedly with pixels in a variety of contexts, including in pixel arrays and with various process technologies and layout variations.

Through this activity, a detailed understanding of the pixel's operation and the requirements of isolating a pixel response, even in an array that provides several parasitics, was obtained. This became an essential piece of knowledge for future designs.

As stated, several chips have been fabricated and tested for identification of sources of error originating from component mismatch. I have characterized and reported critical mismatch statistics for an array of pixels. The standard deviation of the light-to-current transducer was within 1.6% of its mean and the multiplier constant's standard deviation was found to be within 1% of the mean [6,37]. An interesting result of this work was the exclusion of transistor variations as the cause of edges effects, as other studies would suggest. Building an understanding of such effects can be important in pixel arrays where the edges cannot be excluded. Overall, these statistics are essential for understanding limitations and properly building more complex processing systems.

10.2 Pixel Plane Mismatches, Offsets, and Error-Correction Modeling

The further analysis of the entire pixel array structure led into an understanding of pixel offsets and behaviors in the context of the system. I developed a model for the source of column offsets and mismatches that were detrimental to resulting images. I found that the image column offsets could be primarily attributed to the threshold voltage offsets in the pixel transistor pairs. There are additional contributions from parasitic PN junctions in the pixel-array transistors. Furthermore, I explained that the offsets are a function of the image itself, and cannot be removed by simple, static offset correction or scaling. With this understanding, I developed a modulation-demodulation method to remove the offsets. I then designed an imager with the on-chip circuitry for assisting the removal of the offsets. This included switches immediately before and after the pixel plane to modulate the differential signals. The offset removal requires an additional subtraction, which I showed could be done off-chip with digital subtraction or on-chip with a dual-slope integrator. The results of these efforts confirmed the offset model and the validity of the offset removal technique.

The investigations were also used to create a model for the system for algorithm verification [38].

10.3 New Architectures Enabling Increased Functionality and Performance

With the initial investigation completed, the remaining work focused on the circuit development of a platform for larger imagers. The system and all of the components had to be developed to support higher speeds and capacitances while preserving the dynamic range capabilities of the existing architectures. Two larger resolution imagers, a $1K \times 1K$ and a 256×256 imager were designed, fabricated, and tested. I designed the system for the 256×256 imager to include support for two transforms and changed the architecture to have the ability to perform separable convolutions that do not suffer from block edge effects.

I reengineered nearly all of the components for the new architecture along with the sub-systems designs for the 256×256 transform imager. For those, I performed the majority of the sub-component designs and also led the efforts into the developments of parts done in collaboration with others. The circuitry components include analog memory, parasitic offset removal, computational pixel blocks, low-current sensors, and low-current computational systems. I coordinated and provided the majority effort in the testing of the ICs which includes the first physical world images and full on-chip 2-D DCT computations with offset correction. Results also included a convolution operation configured to perform edge enhancement without the artifacts of block-by-block processing.

10.4 Reduced Parasitic Pixel and Pixel-Plane

An important piece of the larger format development was my redesign of the pixel array to minimize parasitic contributions from deselected pixels. The current diversion method used reduces the aforementioned column offsets while increasing differential-mode-to-common-mode ratio of the output signal. This increases the sensing precision and SNR throughout the computations and in the final ADC conversion. The modulation scheme discussed to

remove unwanted signal components has been successfully implemented on these ICs.

10.5 High-Speed Analog Memory

To support the larger pixel arrays a new compact memory structure with large output drive was developed. The first high-speed analog memory structure design was done in collaboration with Dr. Erhan Ozalevli, who primarily designed the back-end amplifier. We devised the FGPFET selection scheme, while I designed the selection and programming circuitry in addition to performing most of the layout and all testing.

The features of the structure include a simple programming scheme that uses the actual output voltage of the memory during the programming cycle, instead of using a slower current measure which requires a mapping to output voltage. The structure has a convenient linear mapping from programmed charge to resulting output voltage, which greatly simplifies the programming process. I tested this structure in the imager ICs and achieved the bandwidths needed for large imagers, though it has not yet been fully utilized for image acquisitions.

10.6 Wide Range Current Sensing and Processing

When testing the $1K \times 1K$ imager, stability issues in the feedback of the logarithmic sensing amplifiers prevented successful operation. After analyzing the structure, I determined criteria for small signal stability in a logarithmic amplifier and designed a new logarithmic sense amplifier to assure stability. Finding that the power consumption in such a structure is proportional to the dynamic range capability, which can be several orders of magnitude, I devised a circuit to dynamically vary gain to reduce this dependence [39, 40]. In certain ideal cases, this can eliminate the dependence of power consumption on dynamic range, saving orders of magnitude of power.

I designed an analog vector-matrix multiplier using using the dynamic gain amplifiers as a front-end. These amplifiers combine the input current sensing and the VMM drive

so that higher speeds can be obtained. The VMM utilized the source nodes of transistors to convey signals instead of using the gates, as the previous imager did, in order to avoid kappa mismatch which introduces a power law in multiplication. This VMM has been fabricated on the new IC and has been tested performing an on-chip DCT of the sensed image. The infrastructure, including programming circuitry, was developed with Jordan Gray, who also contributed large efforts in testing and majority efforts in studying precise programming of the VMM structure. I contributed to the design of a new VMM with faster programming capabilities led by Jordan Gray and the initiation of simulation models and techniques for floating-gate transistor circuits [41].

Because the output of the VMM is a differential current signal, a subtraction is required to convert it to a single value. This subtraction can result in very small currents that can be positive or negative. When this current is fed to a logarithmic amplifier, it flows through feedback transistors that determine I-V conversion characteristic. If the current is near zero, the transistors essentially turn off and this results in slow operation. So, I devised a concept for a bi-directional, compressive transimpedance amplifier with dynamic gain control as a follow-up to the single-ended design. This structure resolves the speed problems by using feedback to reduce input impedance while using two voltage-level shifters in the feedback to guarantee a minimum current flow through the feedback transistors. I worked with Dr. David Abramson to implement the concept and he contributed the primary testing efforts. He has since created design improvements which are included on the latest imager designs.

10.7 Physical System Implementation and Applications

I have engineered a hardware system to test the 256×256 imager, which includes a PCB design and fabrication, FPGA hardware design which includes a soft-processor, C code development for the FPGA processor, and MATLAB code for computer interfacing. The hardware platform creation was assisted by Jungwon Lee and Scott Koziol, though

I designed the circuits and system design. Jordan Gray has significantly helped write revisions of the software components, with some appreciated contributions from Jungwon Lee. As reported, the entire system enabled full on-chip 2-D DCT computation [42], edge-enhancement, and compressive sensing operations [43] in the analog domain.

REFERENCES

- [1] J. Lee, A. Bandyopadhyay, I. Faik Baskaya, R. Robucci, and P. Hasler, "Image processing system using a programmable transform imager," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 5, pp. v/101–v/104 Vol.5, 18-23 March 2005.
- [2] A. Bandyopadhyay and P. Hasler, "A fully programmable cmos block matrix transform imager architecture," in *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, pp. 189–192, 21-24 Sept. 2003.
- [3] P. Hasler, A. Bandyopadhyay, and P. Smith, "A matrix transform imager allowing high-fill factor," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 3, pp. III–337–III–340 vol.3, 26-29 May 2002.
- [4] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nw/mhz, 128/spl times/32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*, pp. 651–654, 3-6 Oct. 2004.
- [5] J. Glossner, K. Chirca, M. Schulte, H. Wang, N. Nasimzada, D. Har, S. Wang, J. Hoane, A.J., G. Nacer, M. Moudgill, and S. Vassiliadis, "Sandblaster low power dsp [parallel dsp arithmetic microarchitecture]," in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*, pp. 575–581, 3-6 Oct. 2004.
- [6] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "Matia: a programmable 80 uw/frame cmos block matrix transform imager architecture," *Solid-State Circuits, IEEE Journal of*, vol. 41, pp. 663–672, March 2006.
- [7] T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9- μ m, 150-mhz, 10-mw, 4 mm 2-d discrete cosine transform core processor with variable threshold-voltage (vt) scheme," *Solid-State Circuits, IEEE Journal of*, vol. 31, pp. 1770–1779, Nov. 1996.
- [8] R. Sarpeshkar, *Efficient precise computation with noisy components: extrapolating from an electronic cochlea to the brain*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997.
- [9] O. Yadid-Pecht, R. Ginosar, and Y. Shacham-Diamand, "A random access photodiode array for intelligent image capture," *Electron Devices, IEEE Transactions on*, vol. 38, pp. 1772–1780, Aug. 1991.
- [10] K. F. Brennan, *The Physics of Semiconductors*. Cambridge University Press, 1999.

- [11] T. Swe and K. Yeo, "An accurate photodiode model for dc and high frequency spice circuit simulation," in *Technical Proceedings of the 2001 International Conference on Modeling and Simulation of Microsystems*, pp. 362 – 365, 2001.
- [12] R. Nixon, S. Kemeny, B. Pain, C. Staller, and E. Fossum, "256x256 cmos active pixel sensor camera-on-a-chip," *Solid-State Circuits, IEEE Journal of*, vol. 31, pp. 2046–2050, Dec. 1996.
- [13] O. Yadid-Pecht and E. Fossum, "Wide intrascene dynamic range cmos aps using dual sampling," *Electron Devices, IEEE Transactions on*, vol. 44, pp. 1721–1723, Oct. 1997.
- [14] T. Delbruck and C. Mead, "Adaptive photoreceptor with wide dynamic range," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 4, pp. 339–342, May 1994.
- [15] D. Yang, A. Gamal, B. Fowler, and H. Tian, "A 640x512 cmos image sensor with ultrawide dynamic range floating-point pixel-level adc," *Solid-State Circuits, IEEE Journal of*, vol. 34, pp. 1821–1834, Dec. 1999.
- [16] O. Yadid-Pecht, B. Pain, C. Staller, C. Clark, and E. Fossum, "Cmos active pixel sensor star tracker with regional electronic shutter," *Solid-State Circuits, IEEE Journal of*, vol. 32, pp. 285–288, Feb. 1997.
- [17] A. Andreou and K. Boahen, "A 590,000 transistor 48,000 pixel, contrast sensitive, edge enhancing, cmos imager-silicon retina," in *Advanced Research in VLSI, 1995. Proceedings., Sixteenth Conference on*, pp. 225–240, 27-29 March 1995.
- [18] Y. Chi, U. Mallik, E. Choi, M. Clapp, G. Gauwenberghs, and R. Etienne-Cummings, "Cmos pixel-level adc with change detection," in *Circuits and Systems, 2006. IS-CAS 2006. Proceedings. 2006 IEEE International Symposium on*, p. 4pp., 21-24 May 2006.
- [19] P. Hasler, B. A. Minch, and C. Diorio, "Floating-gate devices: They are not just for digital memories anymore," in *IEEE International Symposium on Circuits and Systems*, vol. II, (Orlando, Florida), pp. 399–391, 1999.
- [20] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.
- [21] P. Smith, M. Kucic, and P. Hasler, "Accurate programming of analog floating-gate arrays," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 5, pp. V-489–V-492vol.5, 26-29 May 2002.
- [22] M. J. T. Smith and A. Docef, *A Study Guide for Digital Image Processing*. Scientific Publishers, Inc., 1999.
- [23] B. Pain, C. Sun, C. Wrigley, and G. Yang, "Dynamically reconfigurable vision with high performance cmos active pixel sensors (aps)," in *Sensors, 2002. Proceedings of IEEE*, vol. 1, pp. 21–26vol.1, 12-14 June 2002.

- [24] R. McFadyen and F. Schlereth, "Gain-compensated logarithmic amplifier," in *Solid-State Circuits Conference. Digest of Technical Papers. 1965 IEEE International*, vol. VIII, pp. 110–111, Feb 1965.
- [25] S. Chakrabartty, G. Singh, and G. Cauwenberghs, "Hybrid support vector machine/hidden markov model approach for continuous speech recognition," in *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, vol. 2, pp. 828–831 vol.2, 8-11 Aug. 2000.
- [26] A. Aslam-Siddiqi, W. Brockherde, and B. Hosticka, "A 16x16 nonvolatile programmable analog vector-matrix multiplier," *Solid-State Circuits, IEEE Journal of*, vol. 33, pp. 1502–1509, Oct. 1998.
- [27] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A precision cmos amplifier using floating-gate transistors for offset cancellation," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 2, pp. 280–291, 2007.
- [28] T. Serrano-Gotarredona, B. Linares-Barranco, and A. Andreou, "Very wide range tunable cmos/bipolar current mirrors with voltage clamped input," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 11, pp. 1398 – 407, 1999.
- [29] A. El Gamal, B. Fowler, H. Min, and X. Liu, "Modeling and estimation of fpn components in cmos image sensors," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 3301, pp. 168–77, 1998.
- [30] Z. Kalayjian and A. Andreou, "Mismatch in photodiode and phototransistor arrays," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 4, pp. 121–124, May 2000.
- [31] E. Candès and T. Tao, "Near-optimal signal recovery from random projections and universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, pp. 5406–5245, December 2006.
- [32] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. on Pure and Applied Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [33] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–986, June 2007.
- [34] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, April 2006.
- [35] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *Image Processing, 2006 IEEE International Conference on*, pp. 1273–1276, 8-11 Oct. 2006.
- [36] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27 – 44, 2001/01/.

- [37] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "A 80 uw/frame 104x128 cmos imager front end for jpeg compression," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 5318–5321 Vol.5, 23–26 May 2005.
- [38] T. Lee, L. K. Chiu, D. V. Anderson, R. Robucci, and P. Hasler, "Rapid algorithm verification for cooperative analog-digital imaging systems," in *Proc. 50th Midwest Symposium on Circuits and Systems MWSCAS 2007*, pp. 1305–1308, 2007.
- [39] A. Basu, R. W. Robucci, and P. E. Hasler, "A low-power, compact, adaptive logarithmic transimpedance amplifier operating over seven decades of current," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 10, pp. 2167–2177, 2007.
- [40] A. Basu, R. Robucci, and P. Hasler, "A low-power, compact, adaptive logarithmic transimpedance amplifier operating over seven decades of current," in *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2007*, pp. 3055–3058, 2007.
- [41] J. Gray, R. Robucci, and P. Hasler, "The design and simulation model of an analog floating-gate computational element for use in large-scale analog reconfigurable systems," in *Proc. 51st Midwest Symposium on Circuits and Systems MWSCAS 2008*, pp. 253–256, 2008.
- [42] R. Robucci, J. Gray, D. Abramson, and P. E. Hasler, "A 256x256 separable transform cmos imager," in *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2008*, pp. 1420–1423, 2008.
- [43] R. Robucci, L. K. Chiu, J. Gray, J. Romberg, P. Hasler, and D. Anderson, "Compressive sensing on a cmos separable transform image sensor," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2008*, pp. 5125–5128, 2008.